

FIG. 1. (prior art)

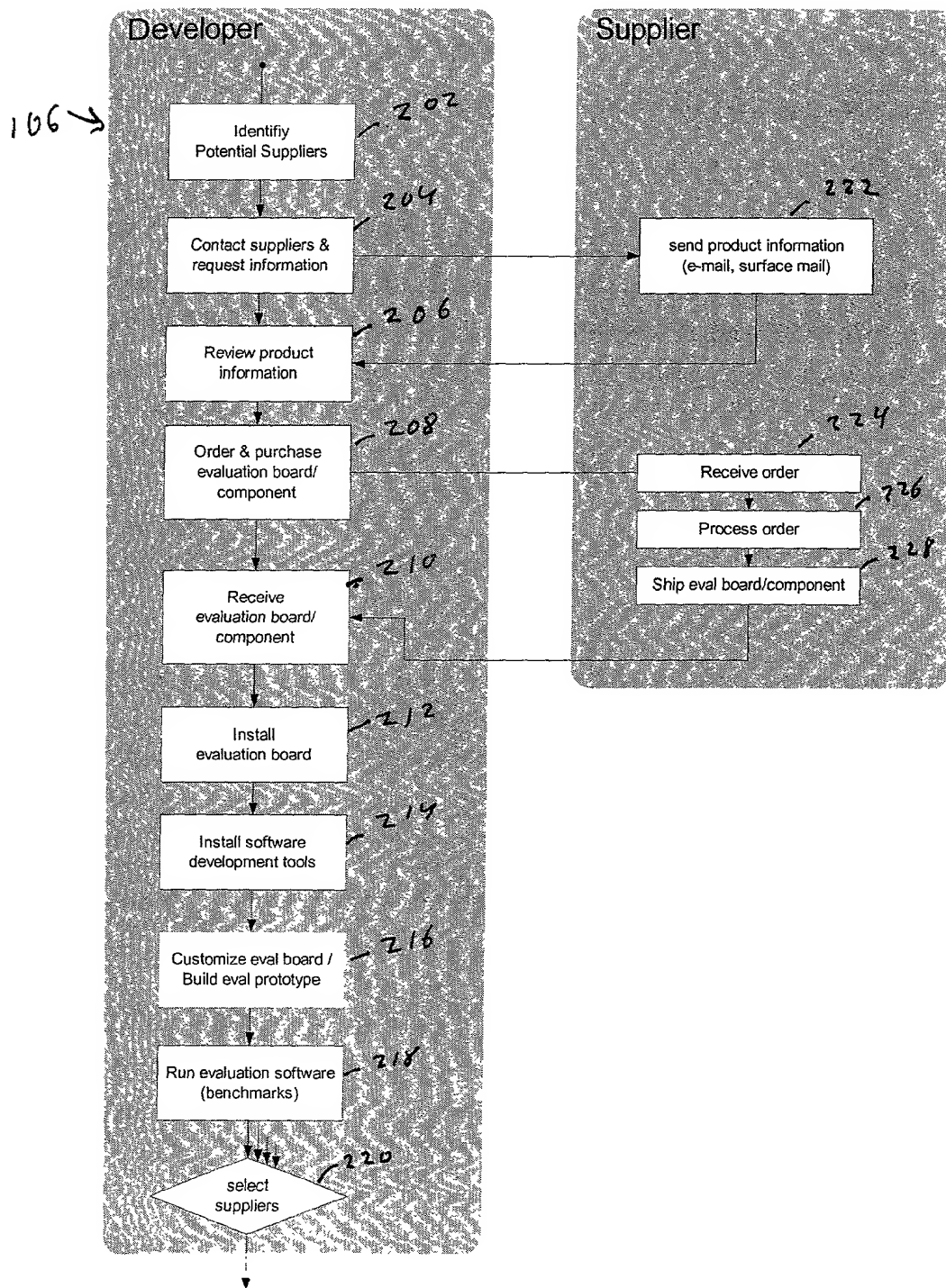


FIG. 2 (prior art)

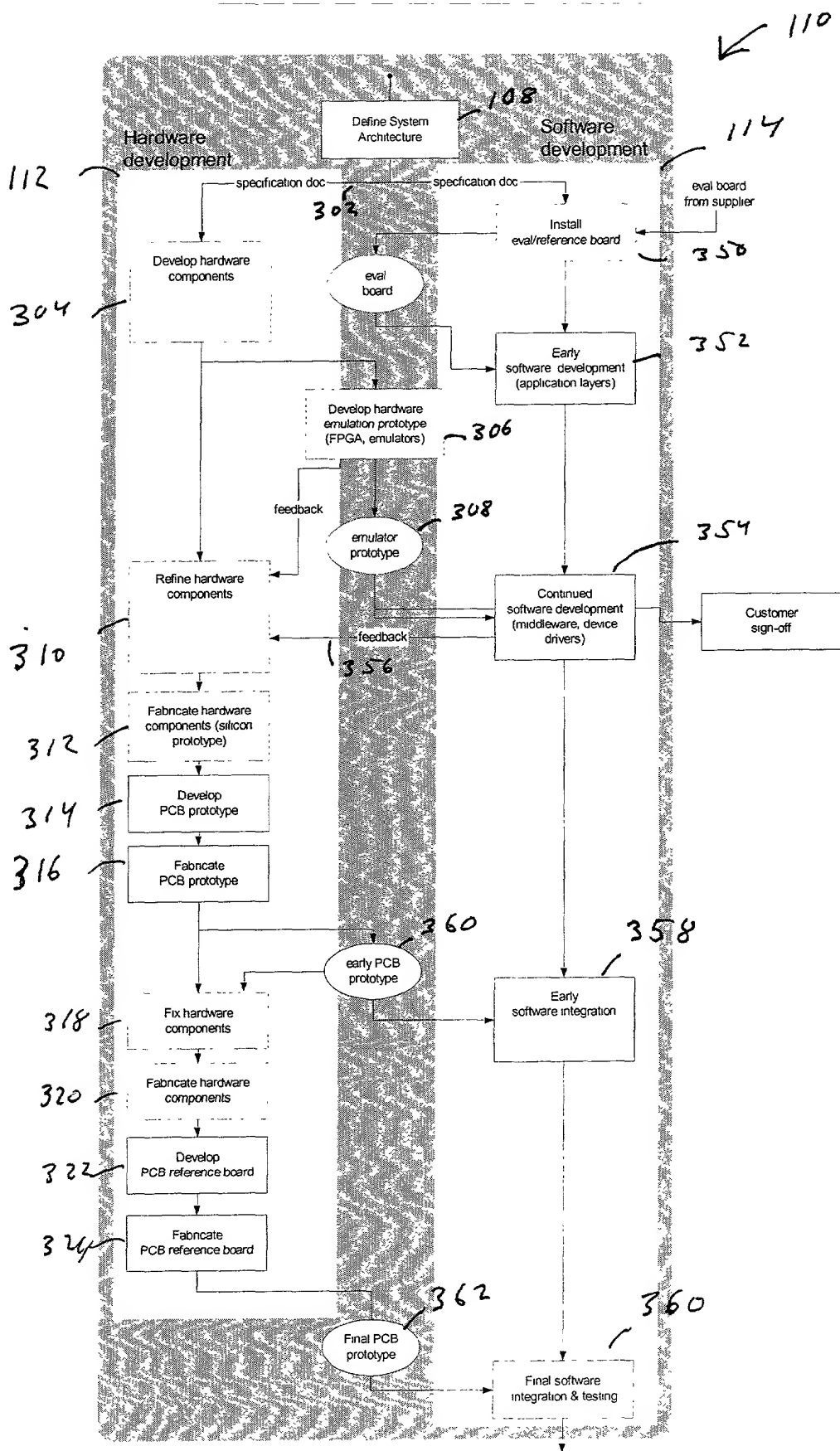


FIG. 3 (prior art)

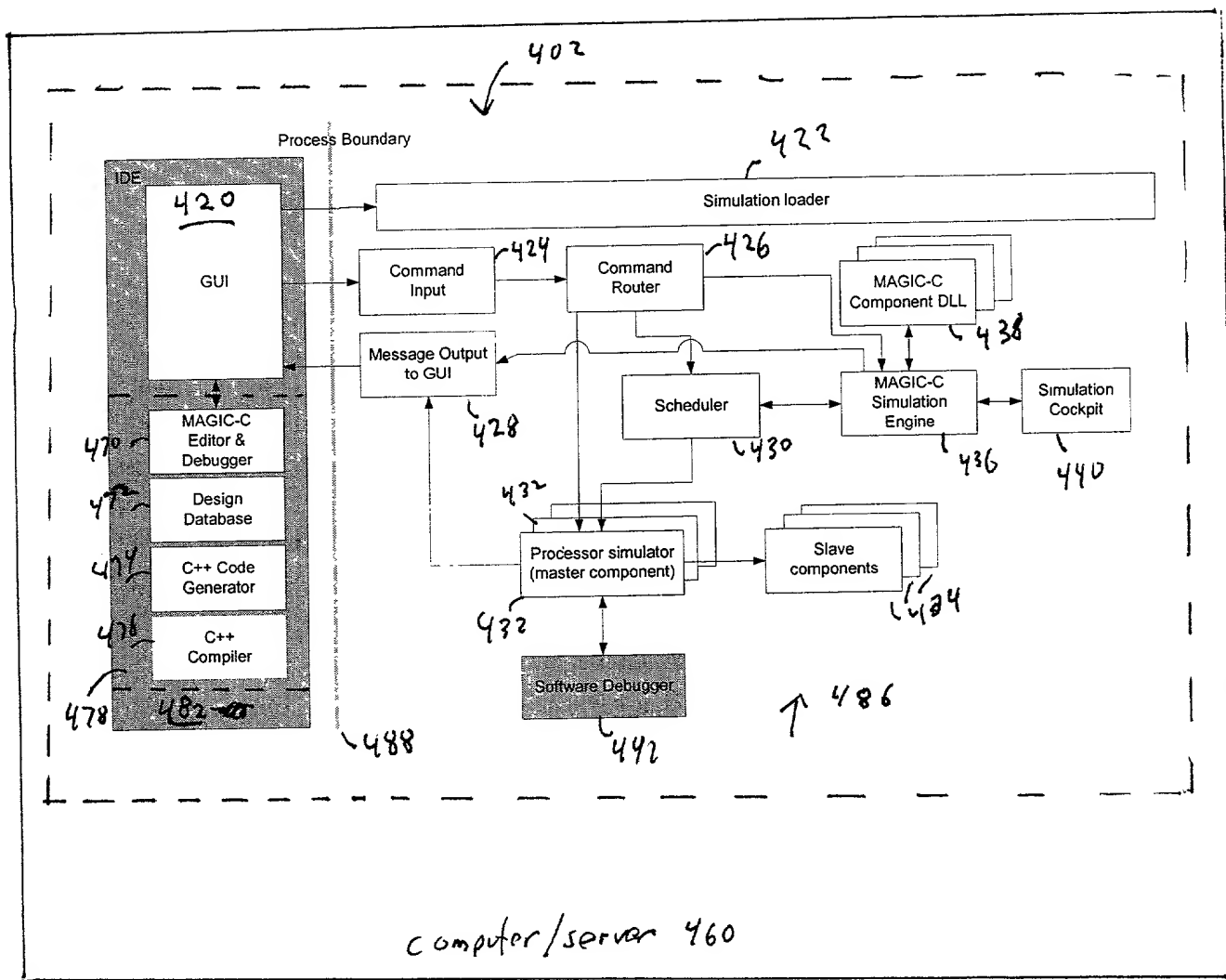


FIG. 4A



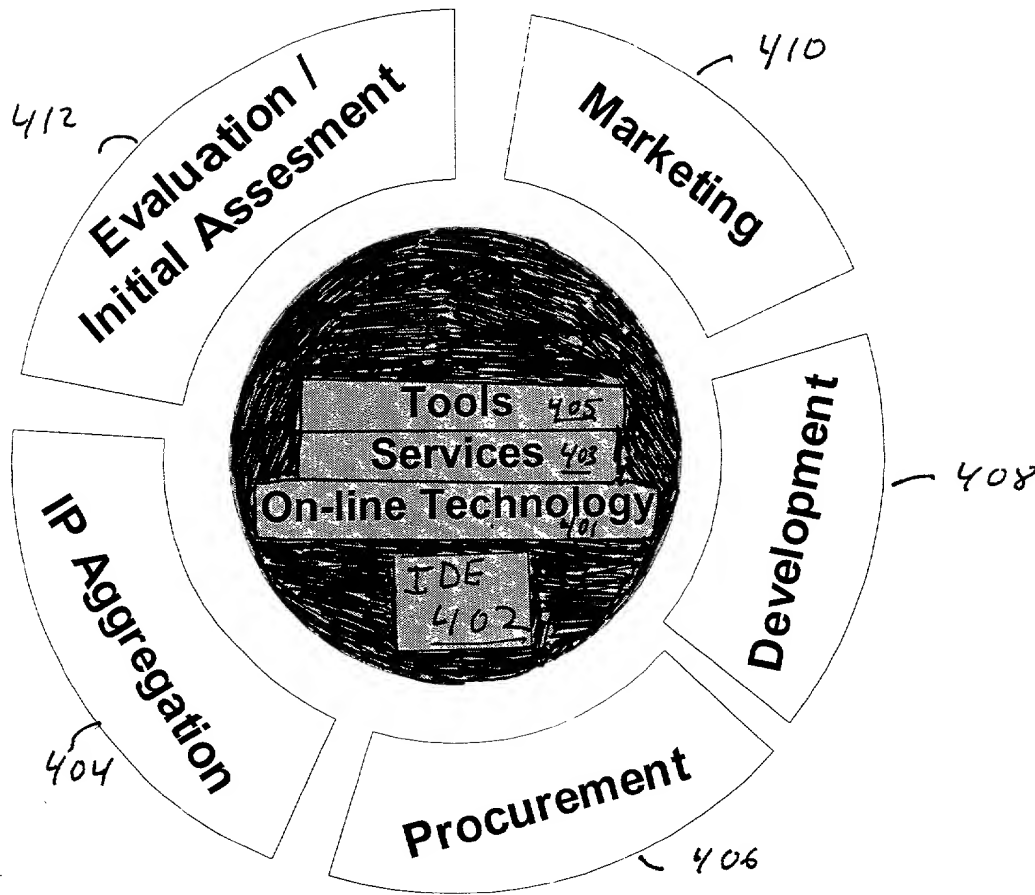


FIG. 4B

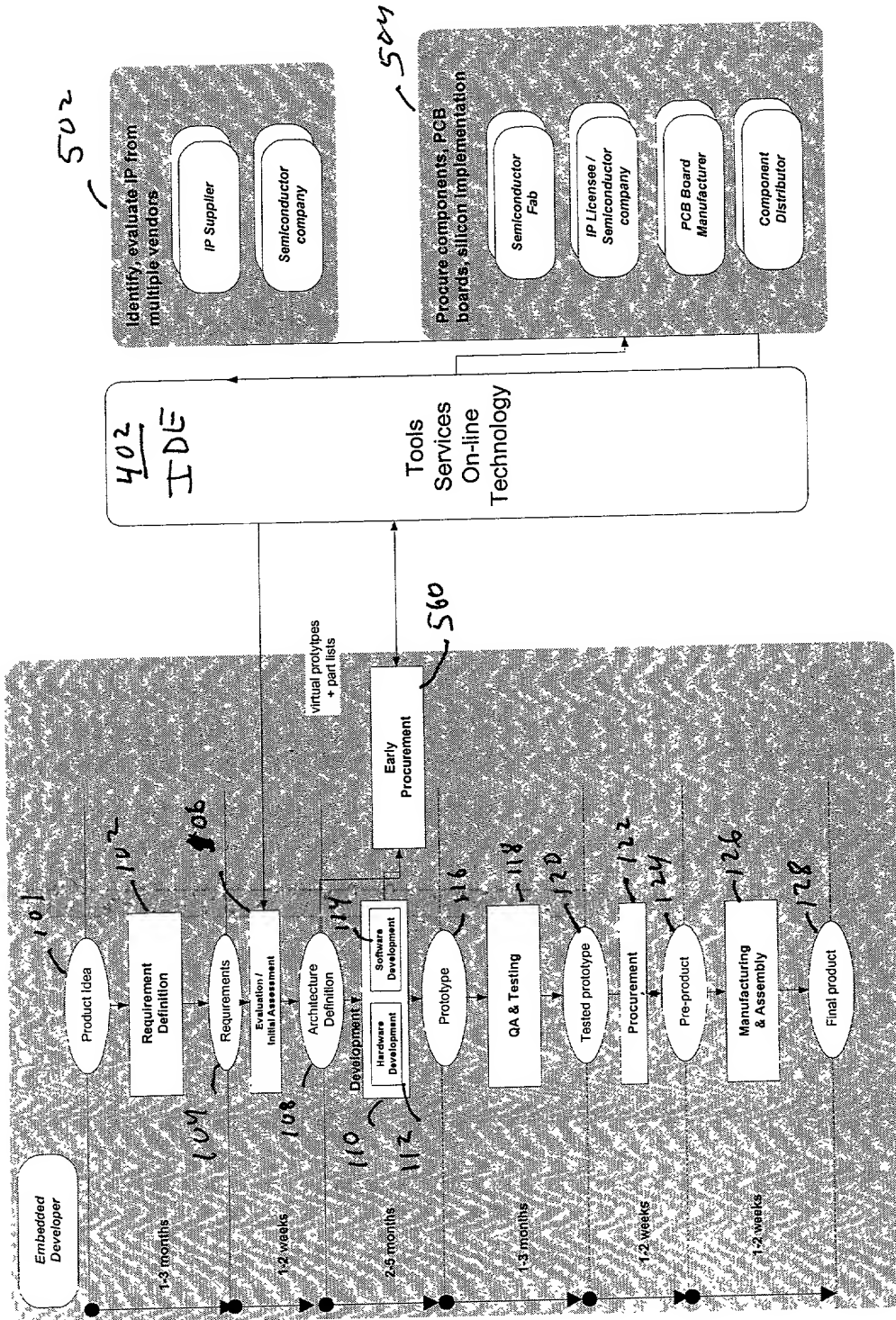


FIG. 5

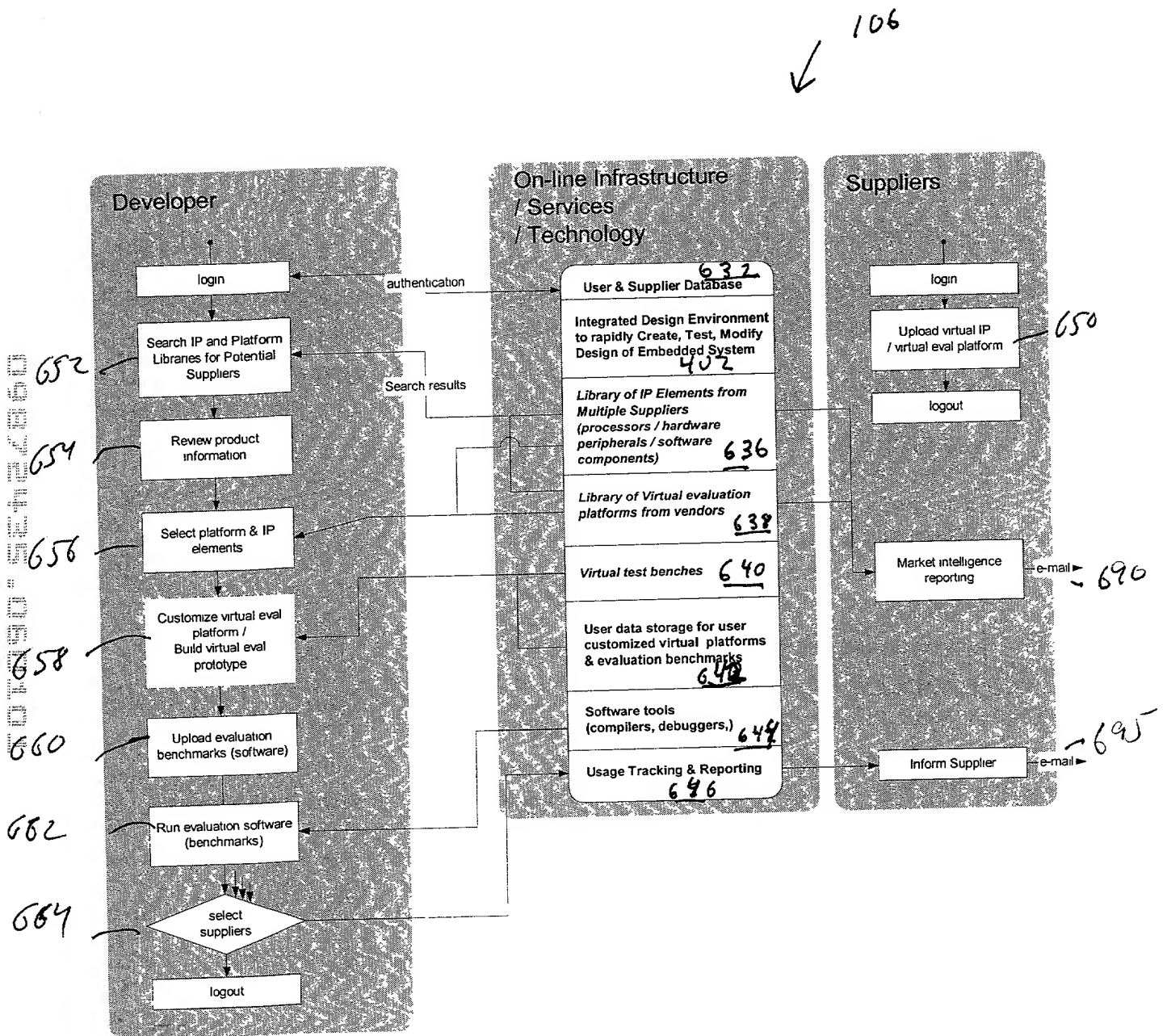


FIG. 6

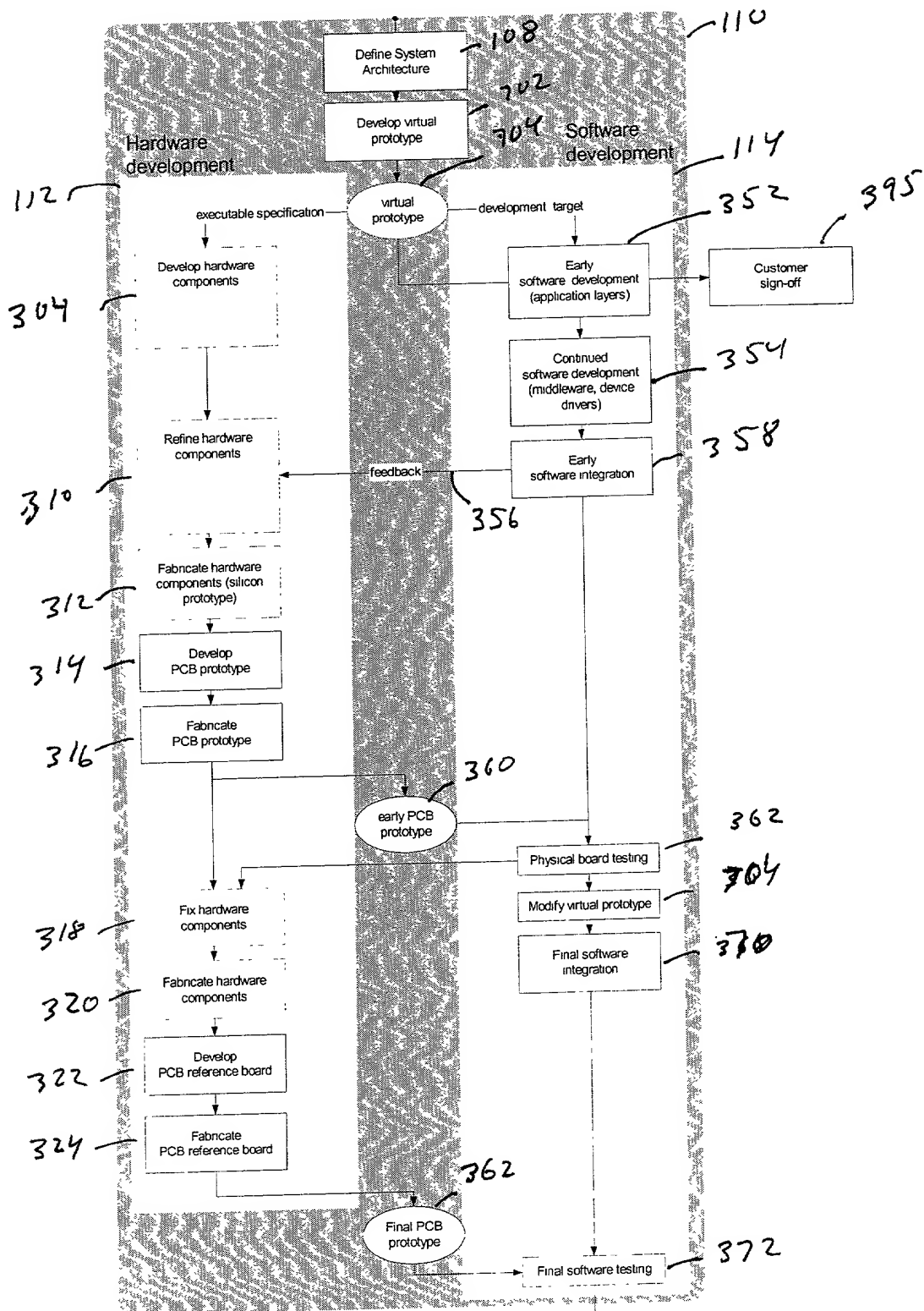


FIG. 7

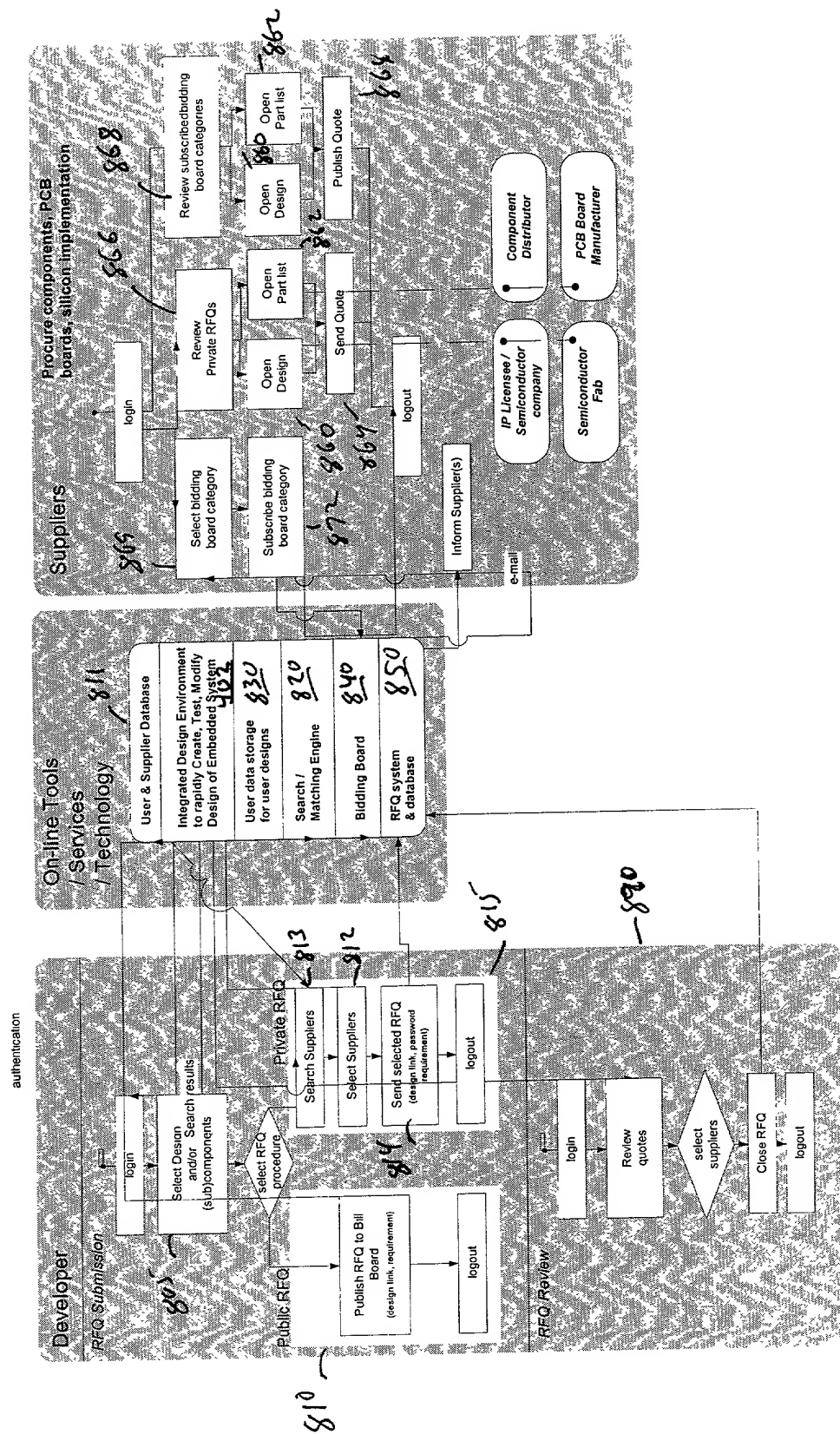


FIG. 8

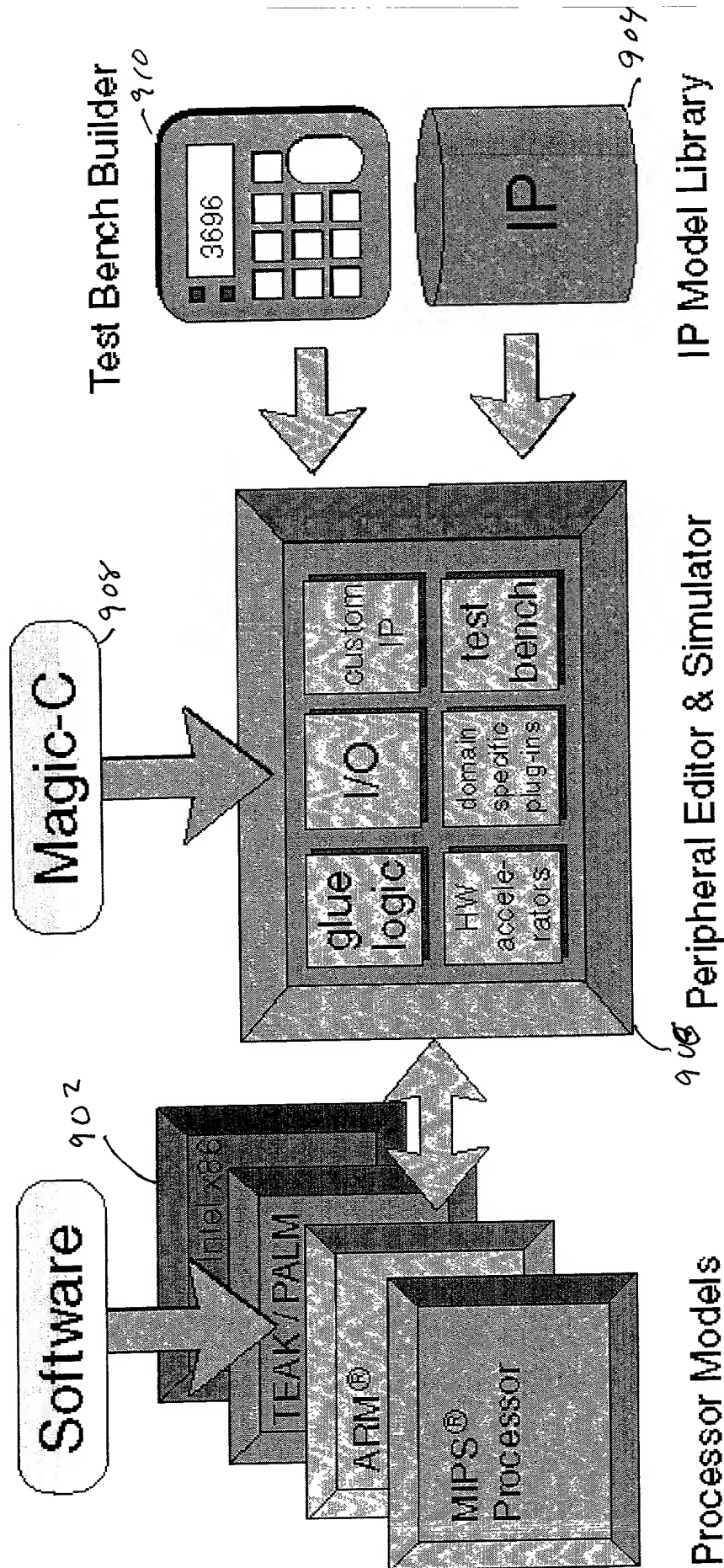


FIG. 9



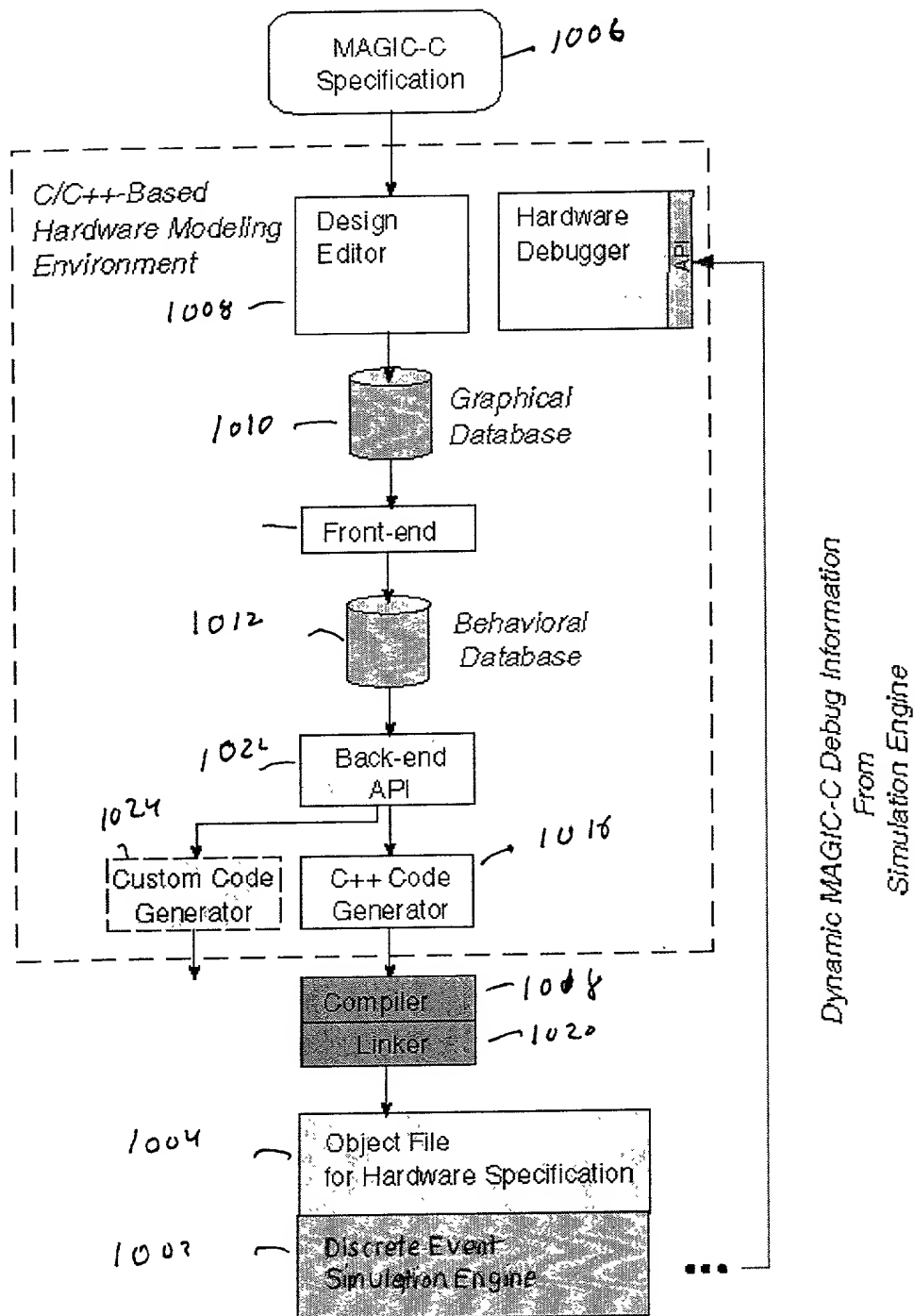


FIG. 10

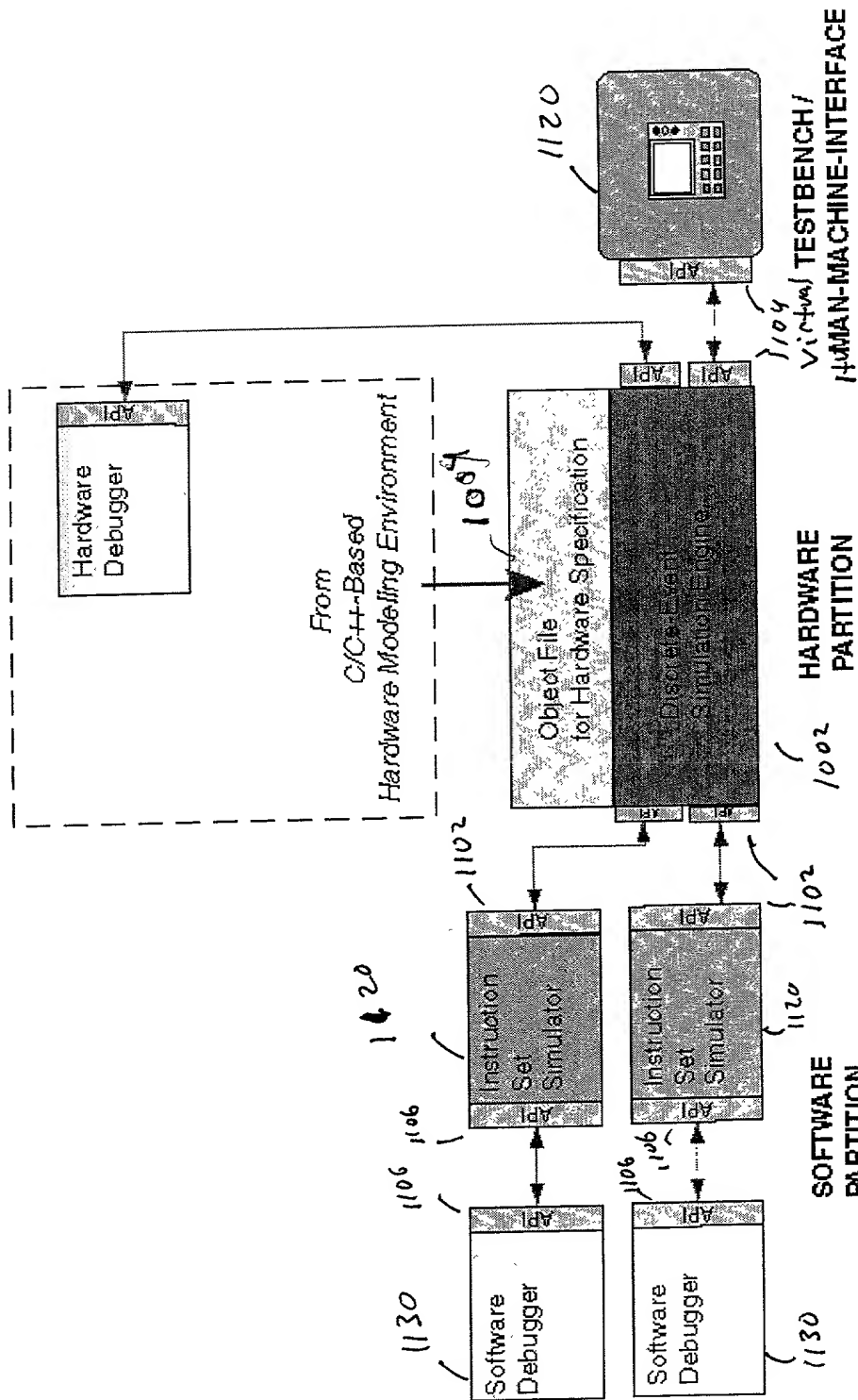


FIG. 11

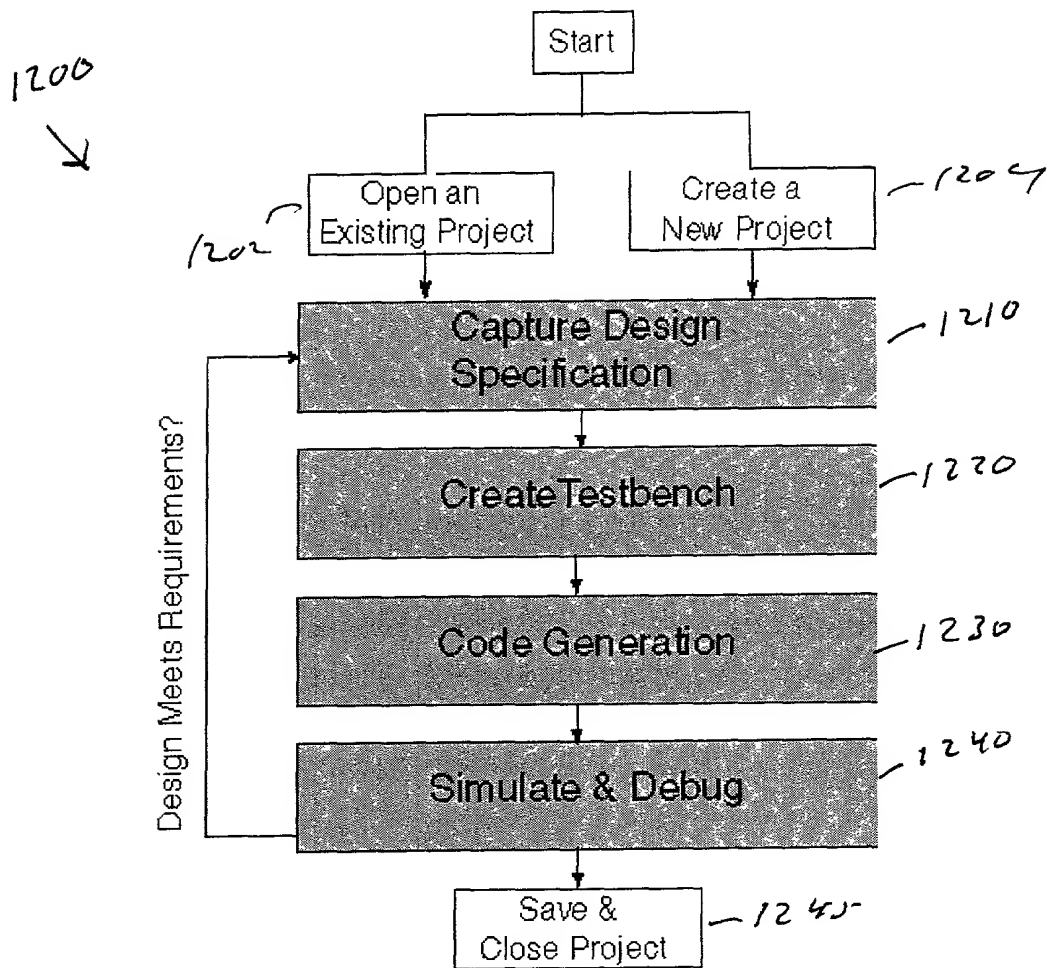


FIG. 12

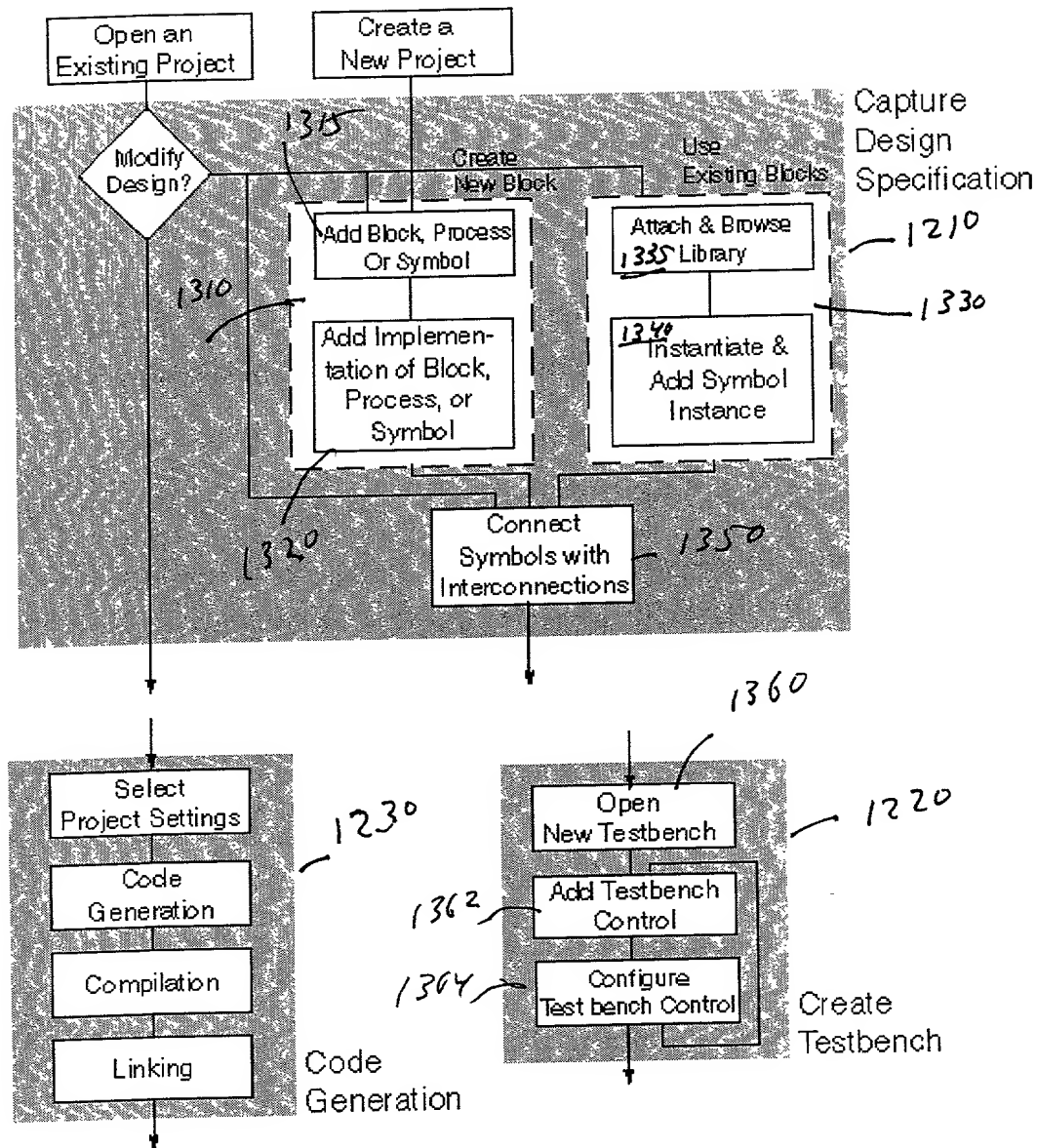


FIG. 13

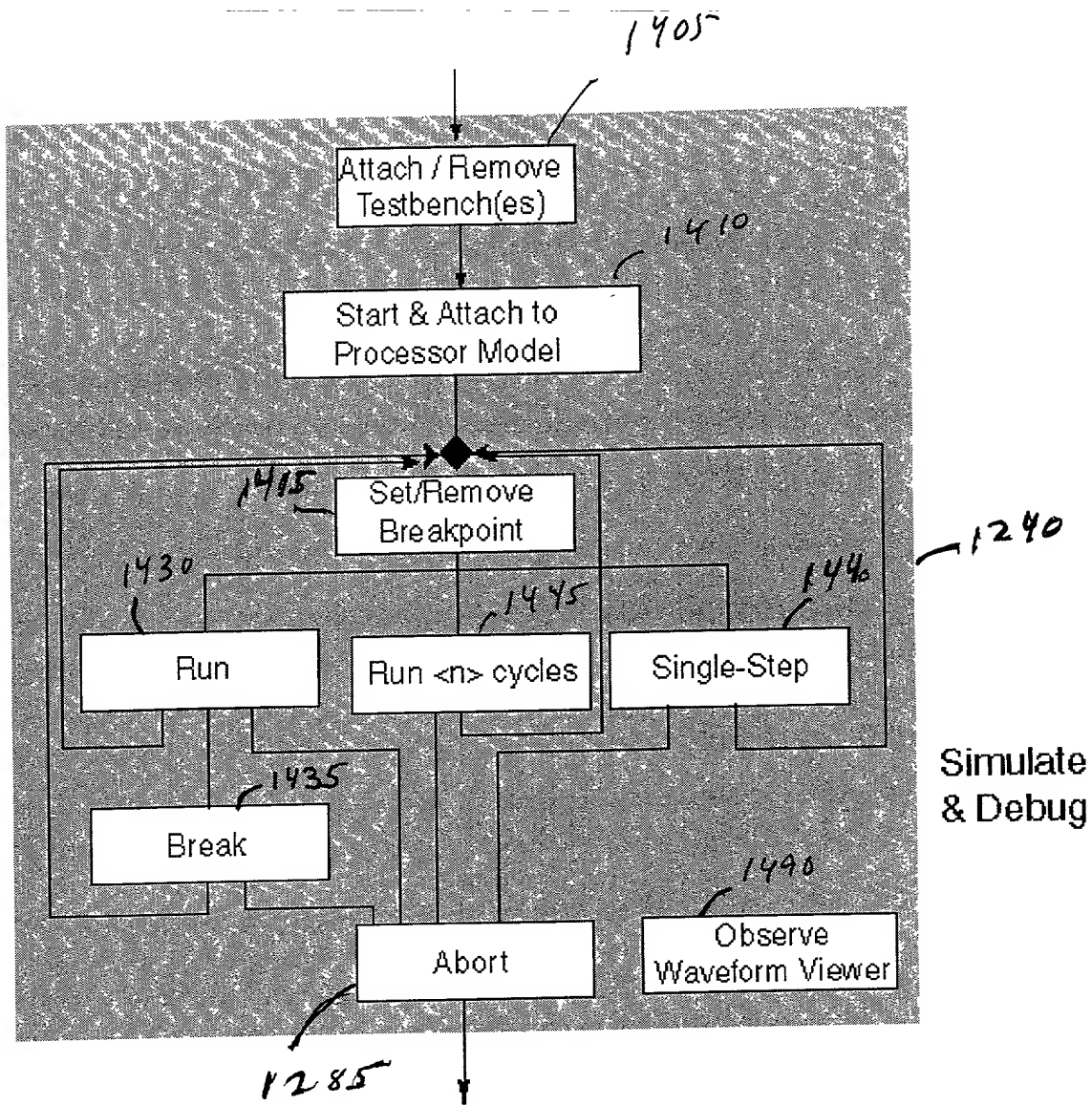


FIG. 14

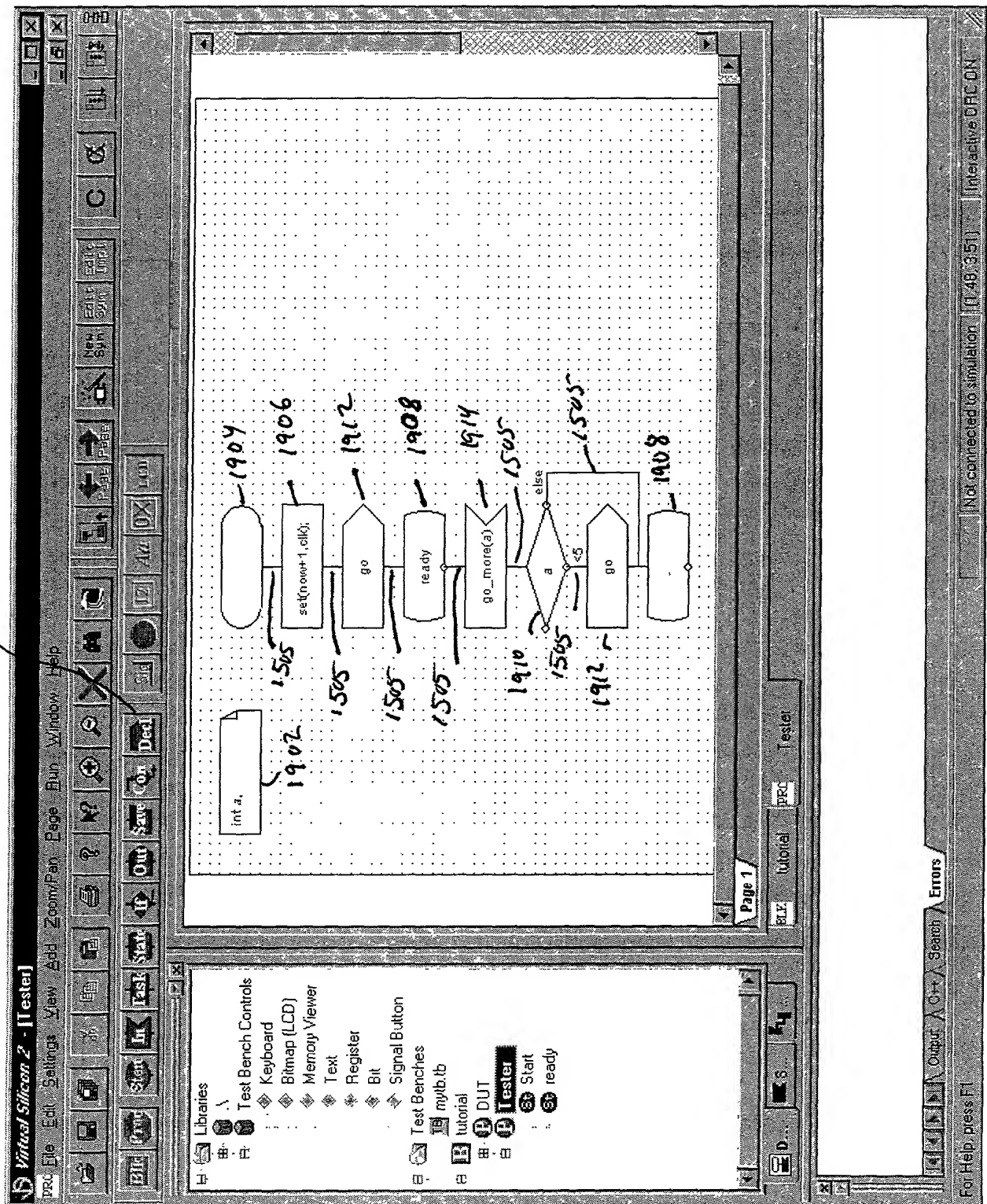


FIG. 15



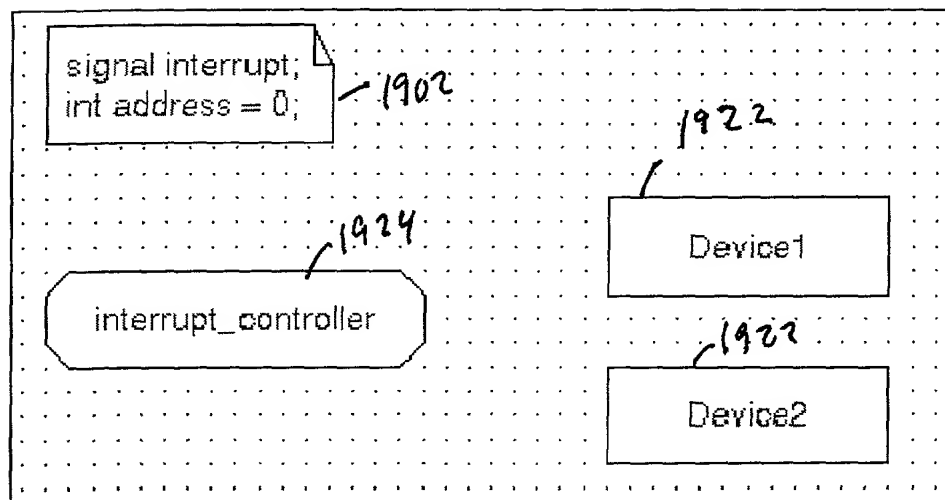


FIG. 16

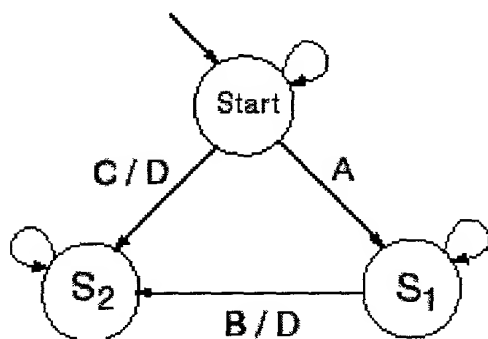


FIG. 17(a)

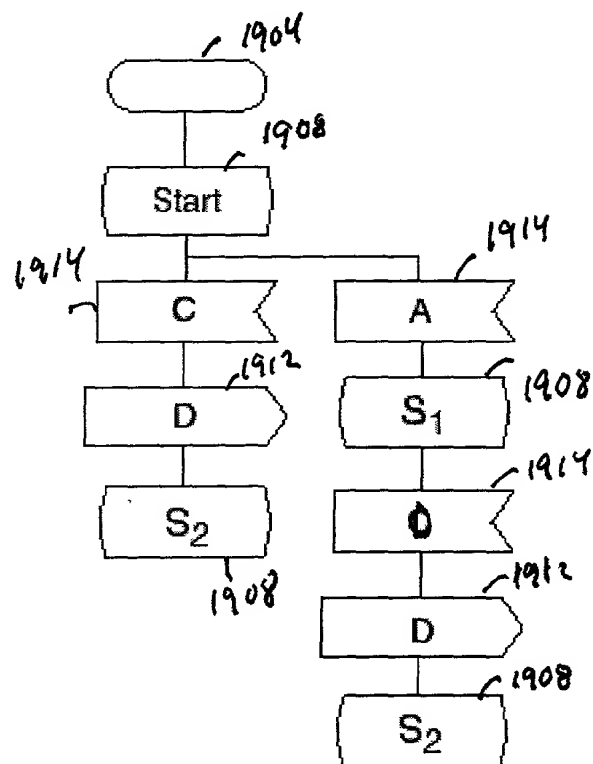
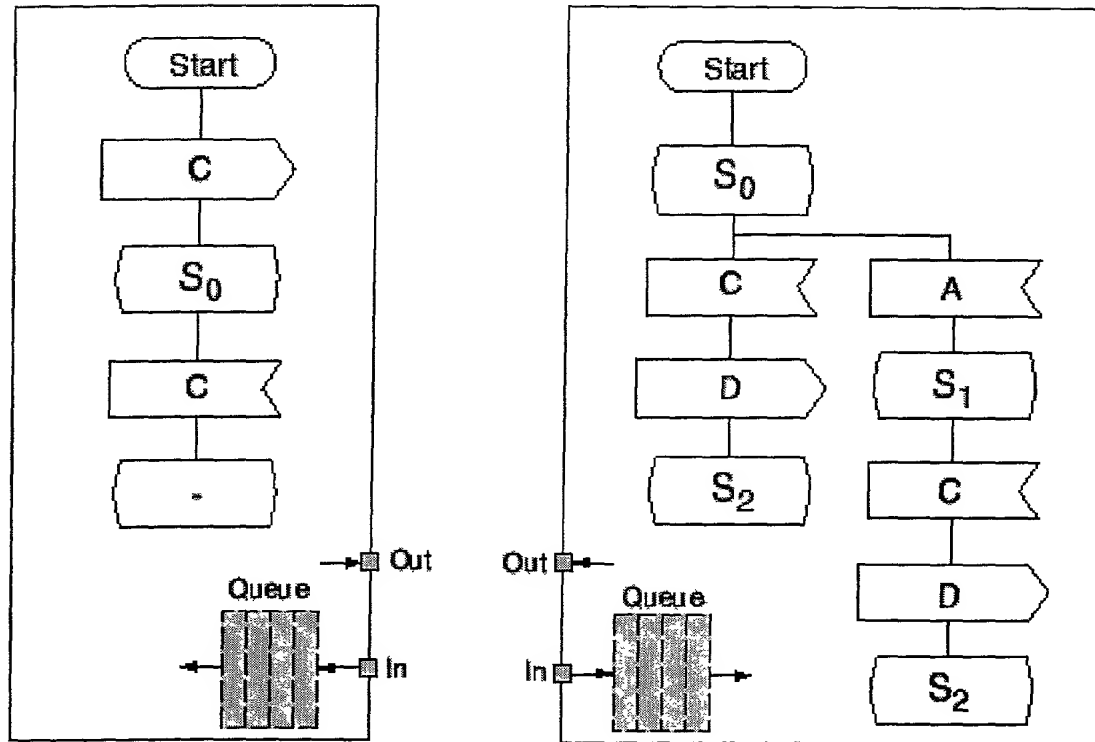


FIG. 17(b)



FJG-18

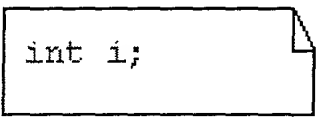
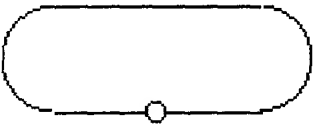
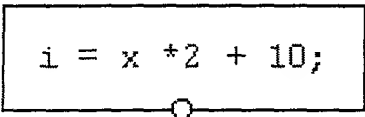
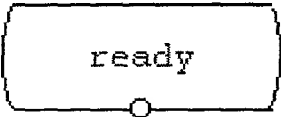

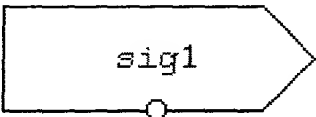
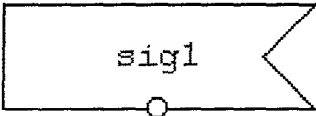
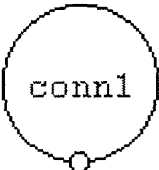
Name	Graphical symbol	Description
Declaration		defines local variables and signals.
Start		Starting point of the Finite State Machine execution at initialization time.
Task		Execution block, containing ANSI-C statements to be executed.
State		Location where FSM waits in until a triggering Signal is received.
Decision		Directs execution flow based on the result of expression evaluation inside the decision construct.
Signal-Out		Sending of a communication signal (with an optional payload).
Signal-In		Receiving of a communication signal (with an optional payload)
Connector		Allows to split designs over multiple pages, and connects the control flow between these different pages.

FIG 19A

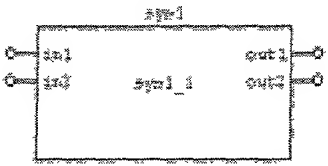


1920 Symbol		Captures design hierarchy and structure. Communication is done through pins on the outline of the symbol. Allows to re-use functional behavior by supporting multiple instances.
1922 Blocks		Captures design hierarchy and structure. Communication is done through signals declared at higher scopes. Communication is done by signal name matching (rather than by pin connection). A block can contain multiple processes.
1924 Process		Acts as leaf node in the design hierarchy, and captures a single FSM. By definition, all processes are concurrent at all times.

FIG. 19B

```

// External interface
extern_signal WR(unsigned int,unsigned int, unsigned int);
extern_signal RD(unsigned int,unsigned int);
extern_signal Write_Ack;
extern_signal Read_Data(unsigned int);
extern_signal CNTINTR(unsigned int);
// Local variables
signal start_clk;
clock clk;
bool clock_started;
unsigned int LIMIT; //write register
VS_int COUNT;
//temp vars
unsigned int data,width, addr;

```

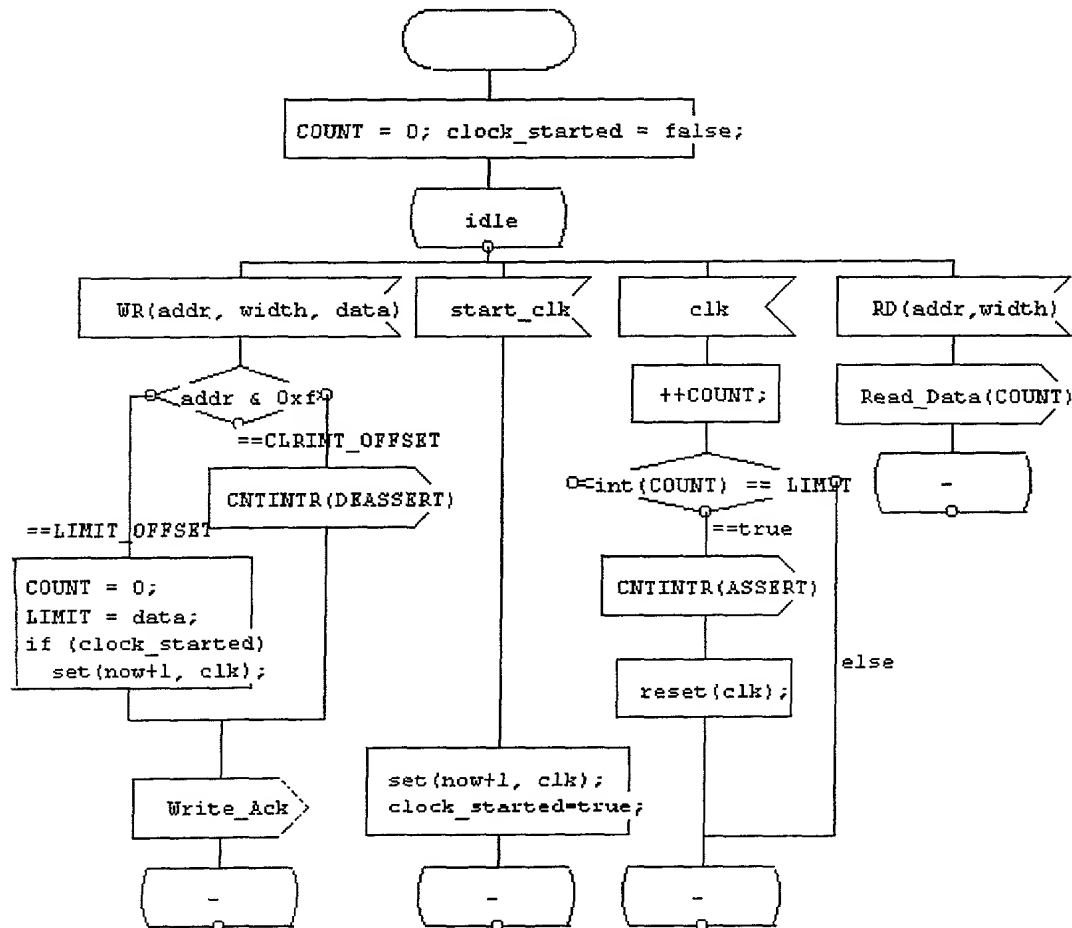


FIG. 20

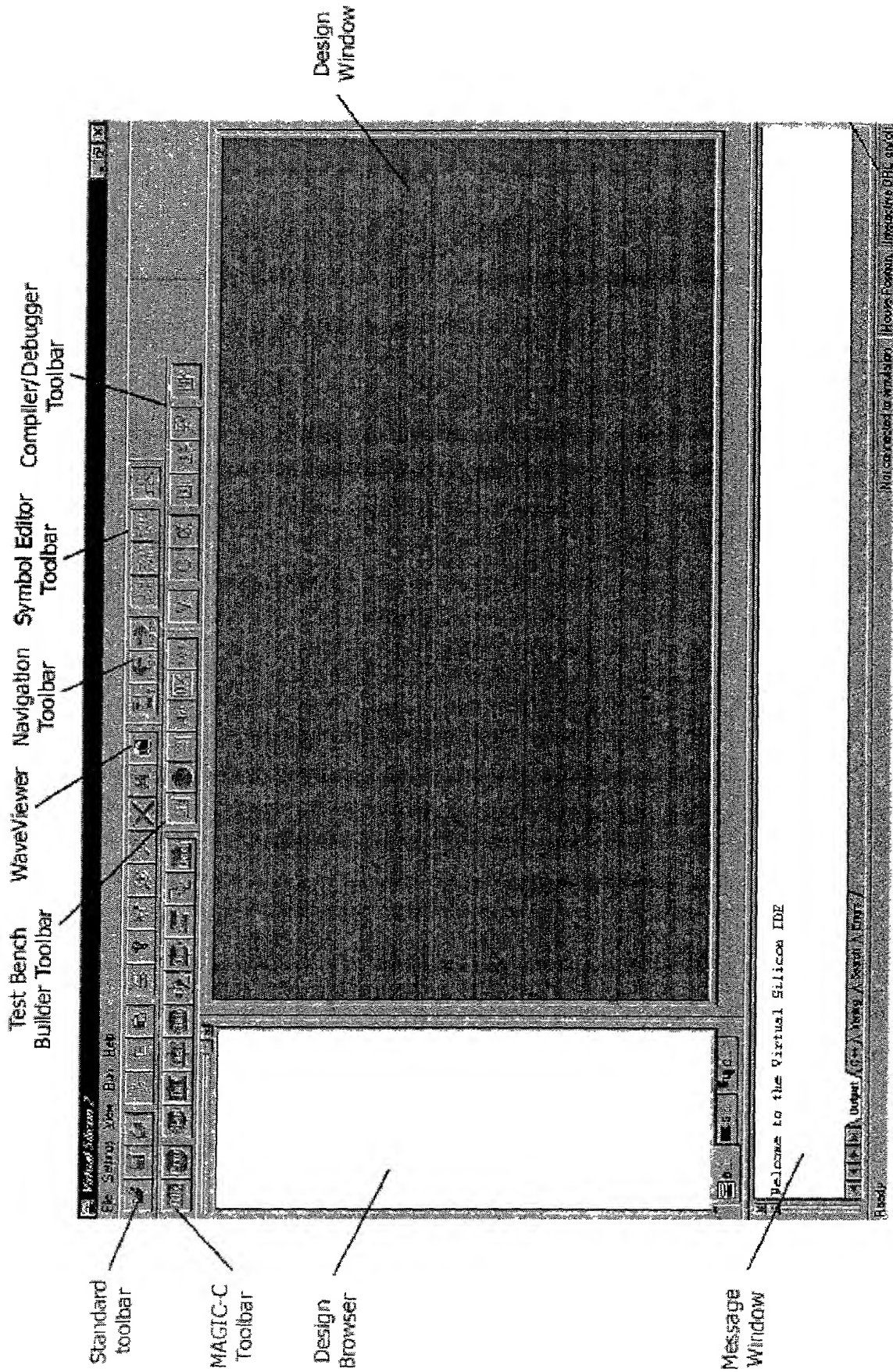


FIG. 21





FIG. 22

Standard Toolbar

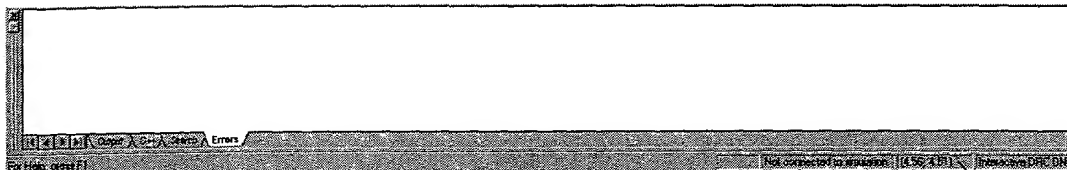
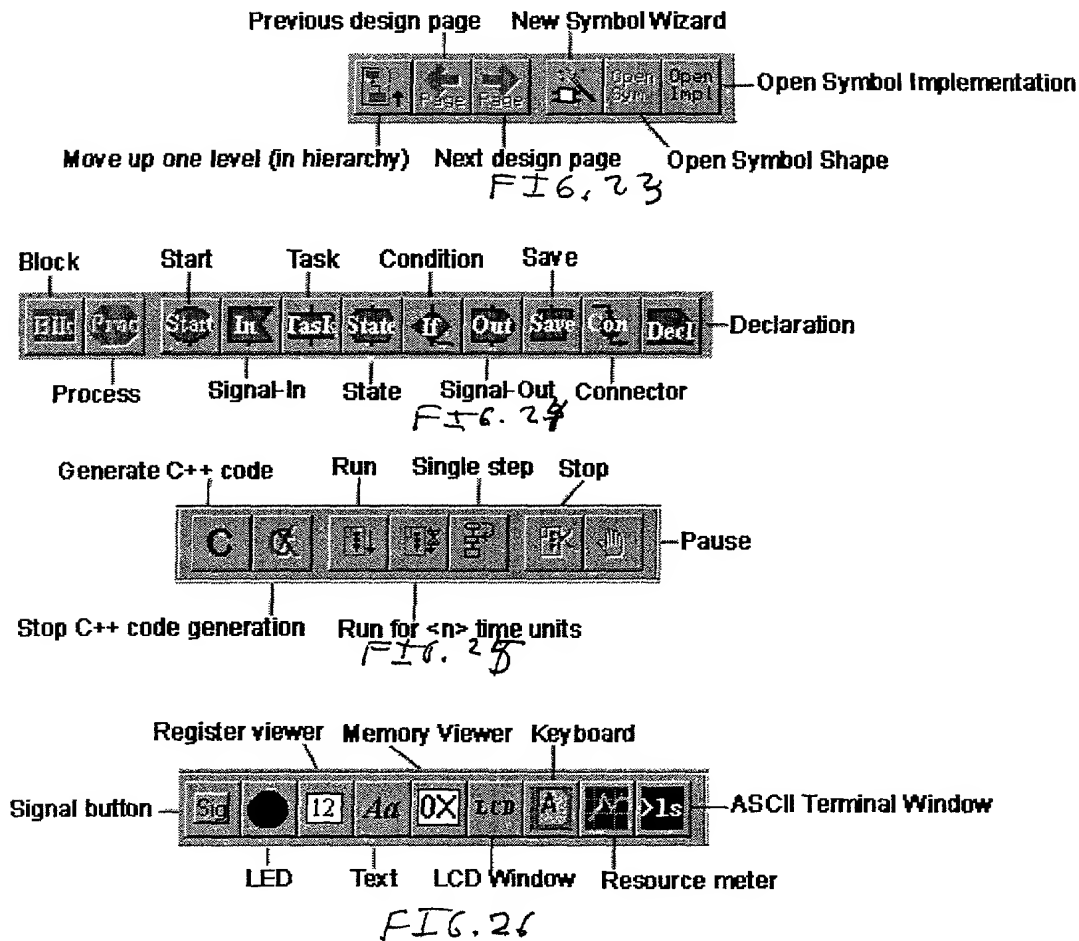


FIG. 23

Status Bar

Message Window, Showing Different Message Tabs, and the Status Bar



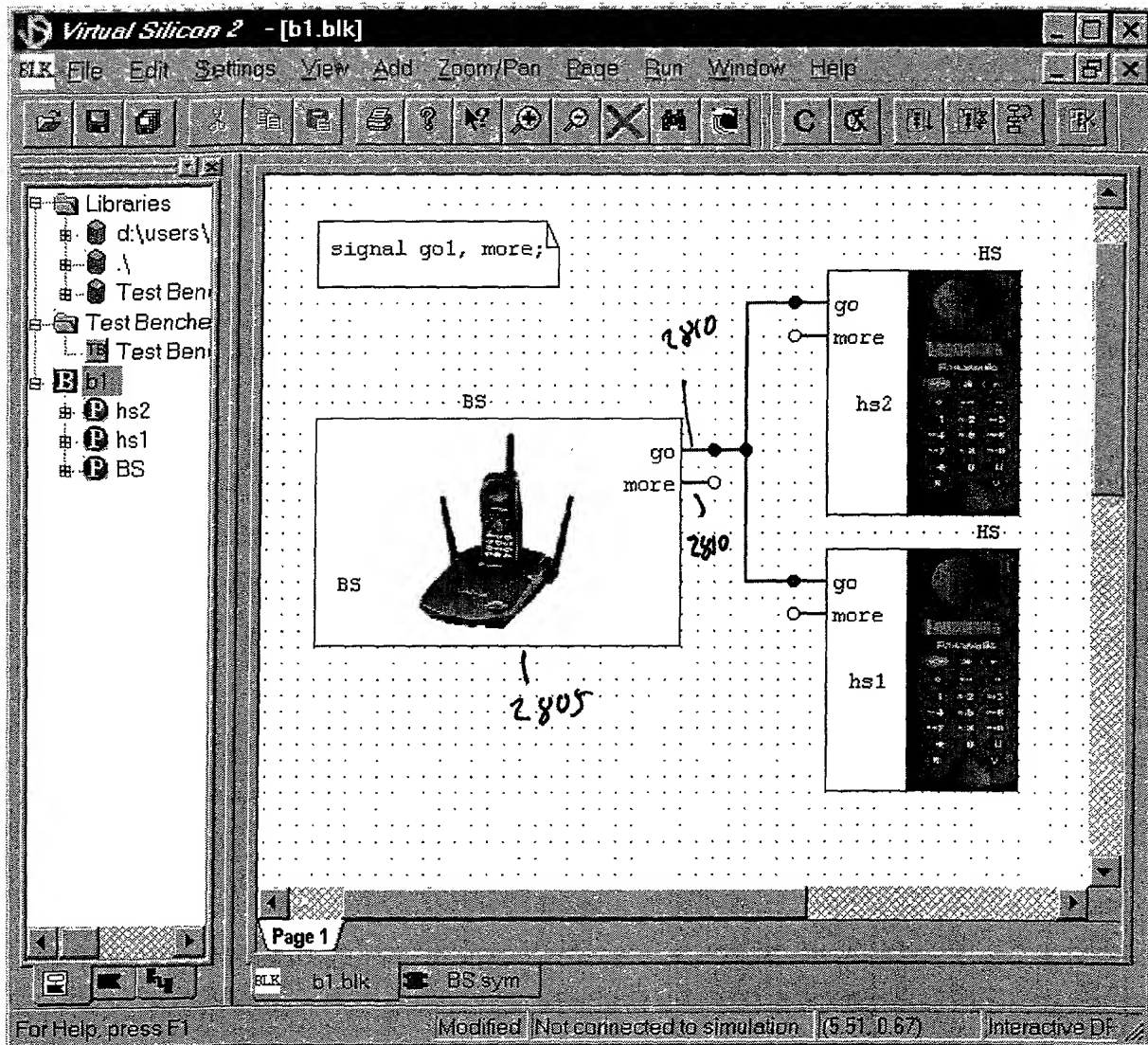


FIG. 28

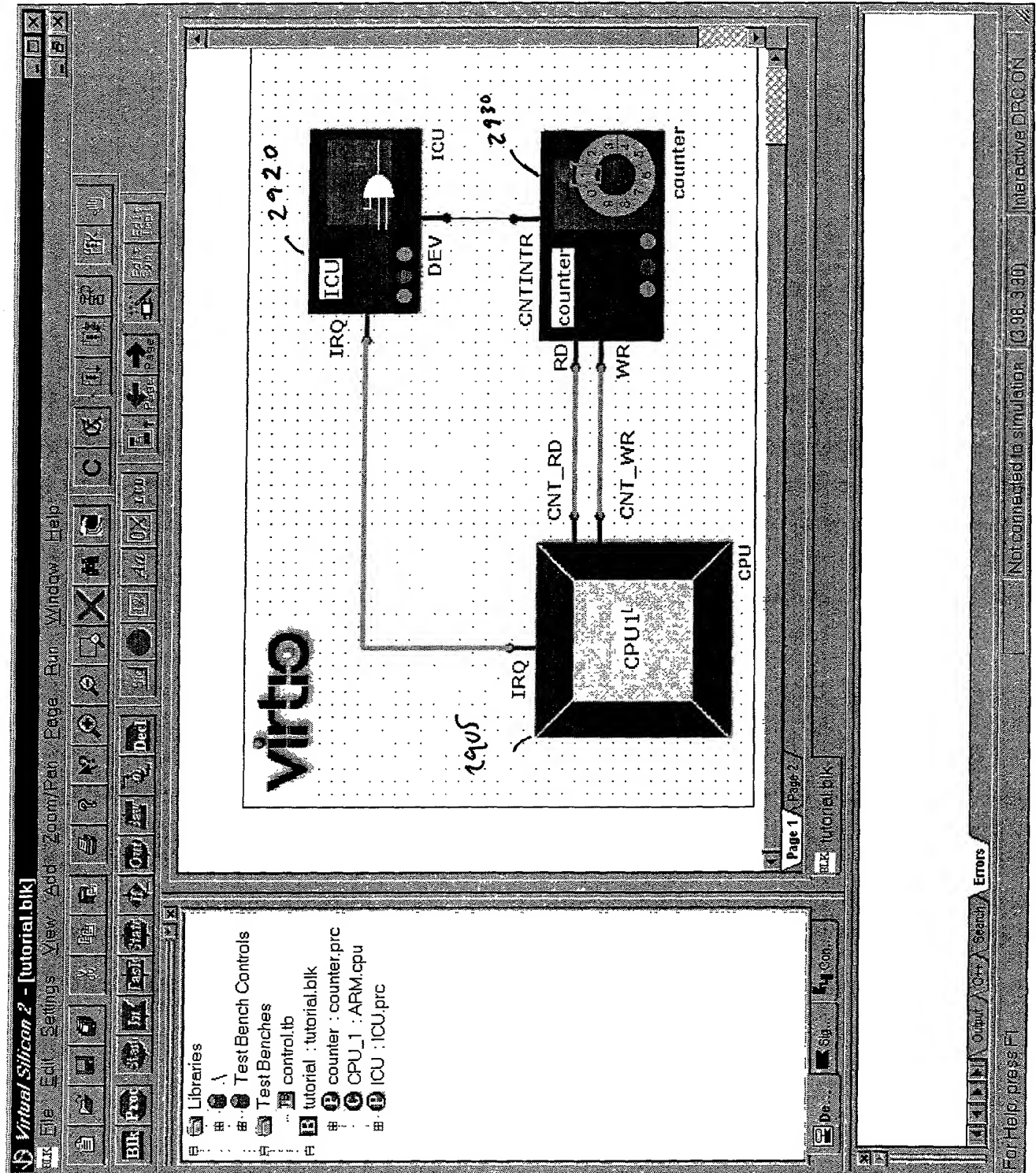
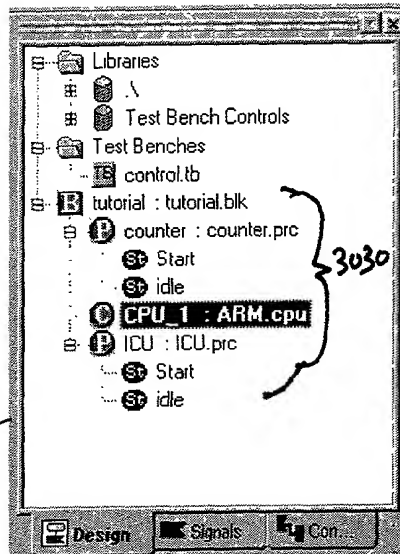


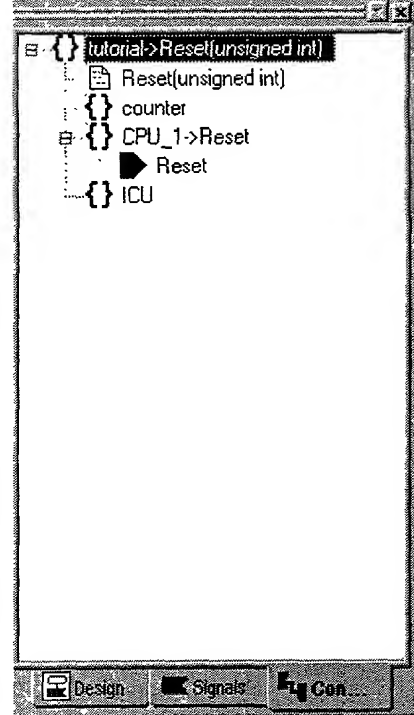
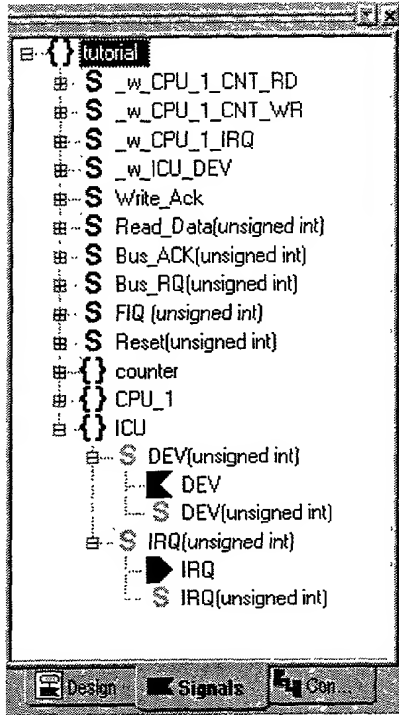
FIG. 29



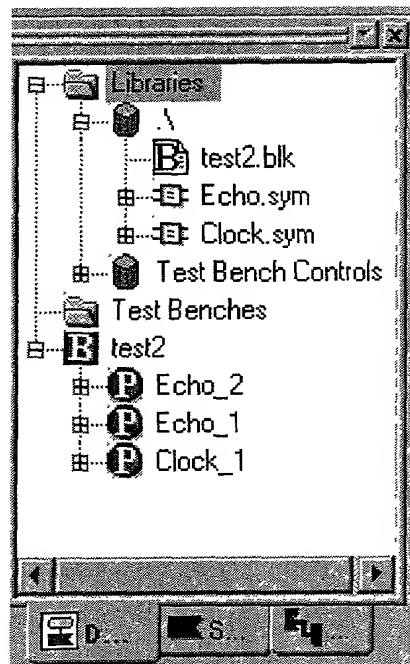
3000

FIG. 30

3020



3030



7

FIG. 31

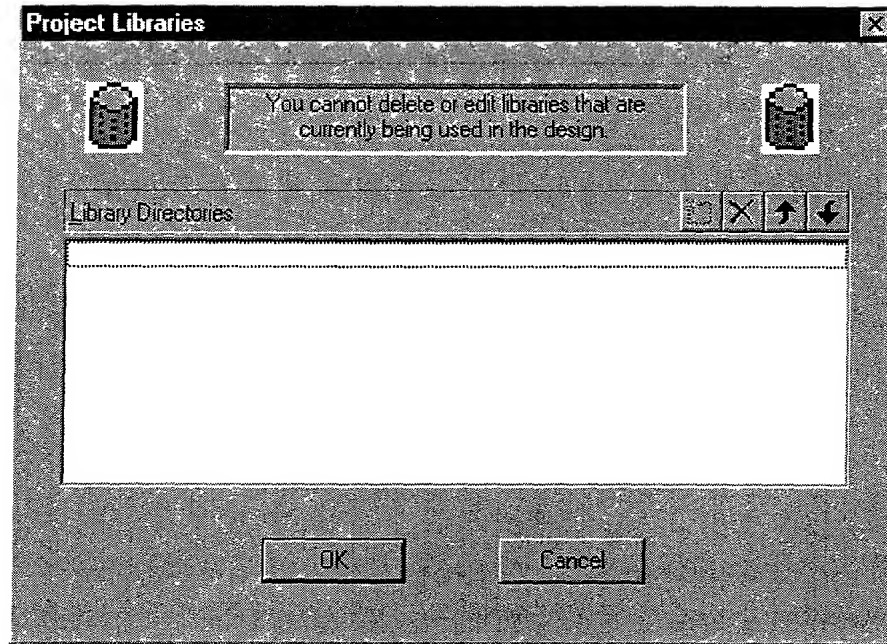
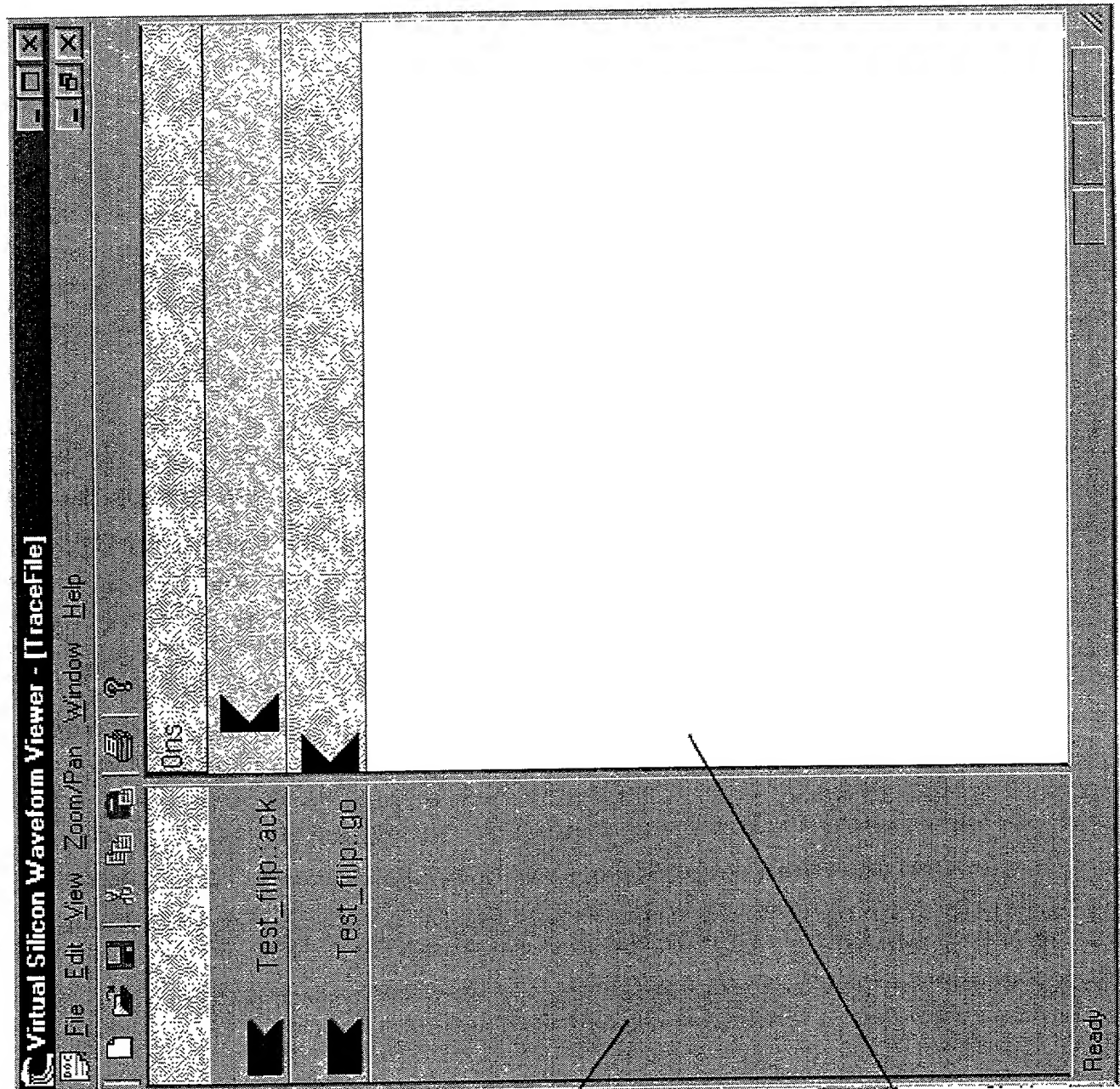


FIG. 32





Name Display  
Window

Waveform  
Display

FIG. 33A

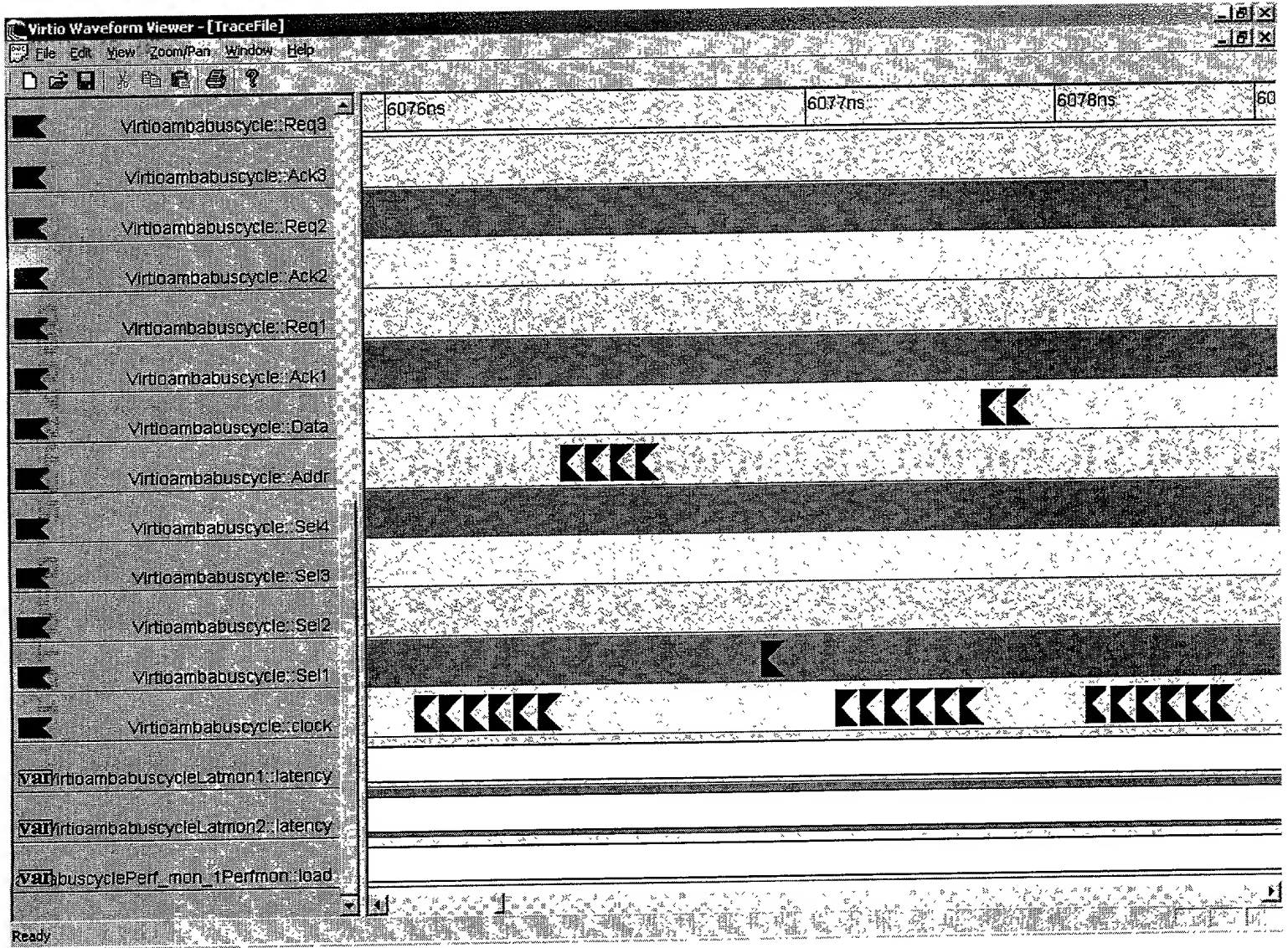


FIG. 33b

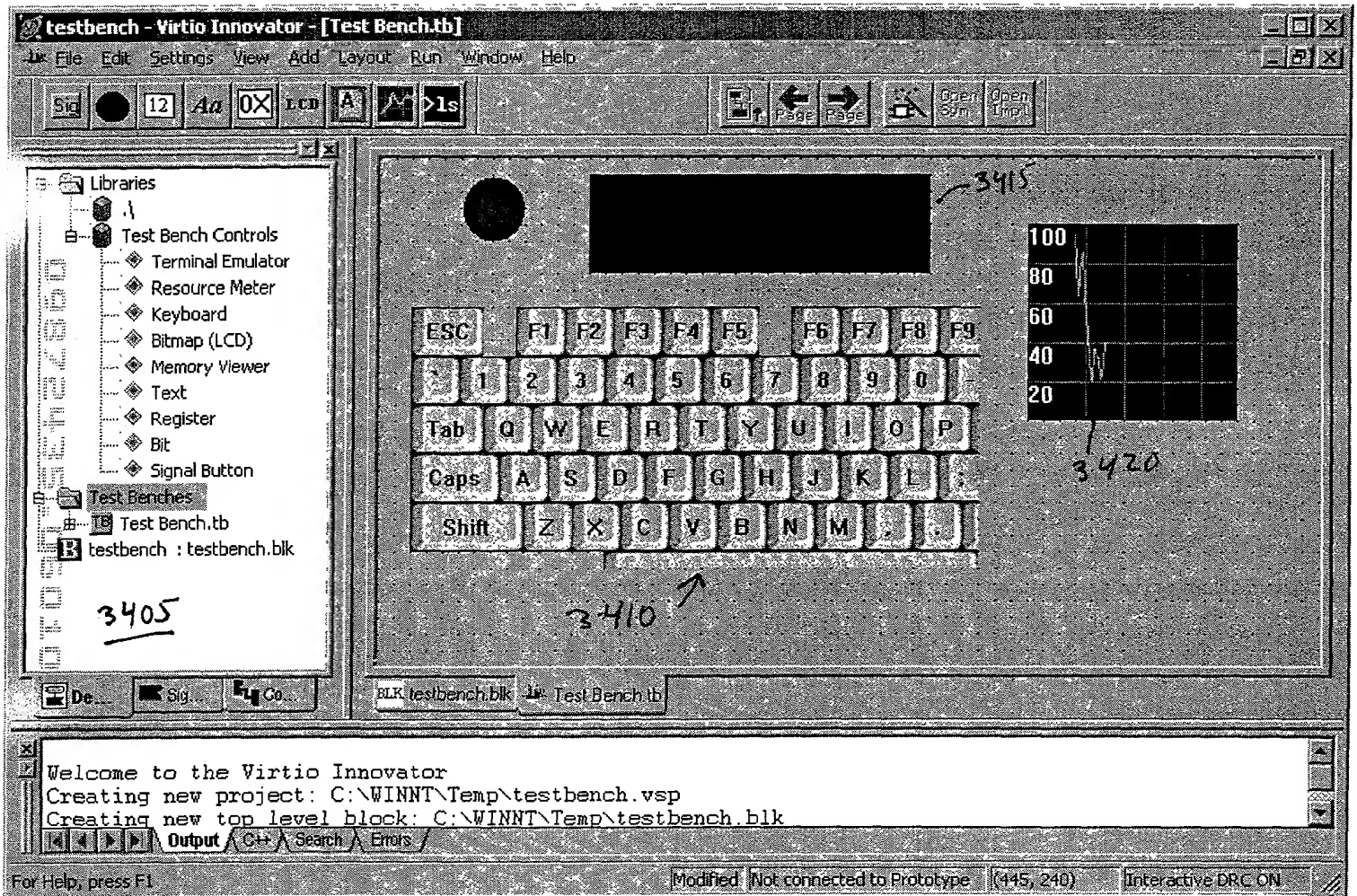


FIG. 34A



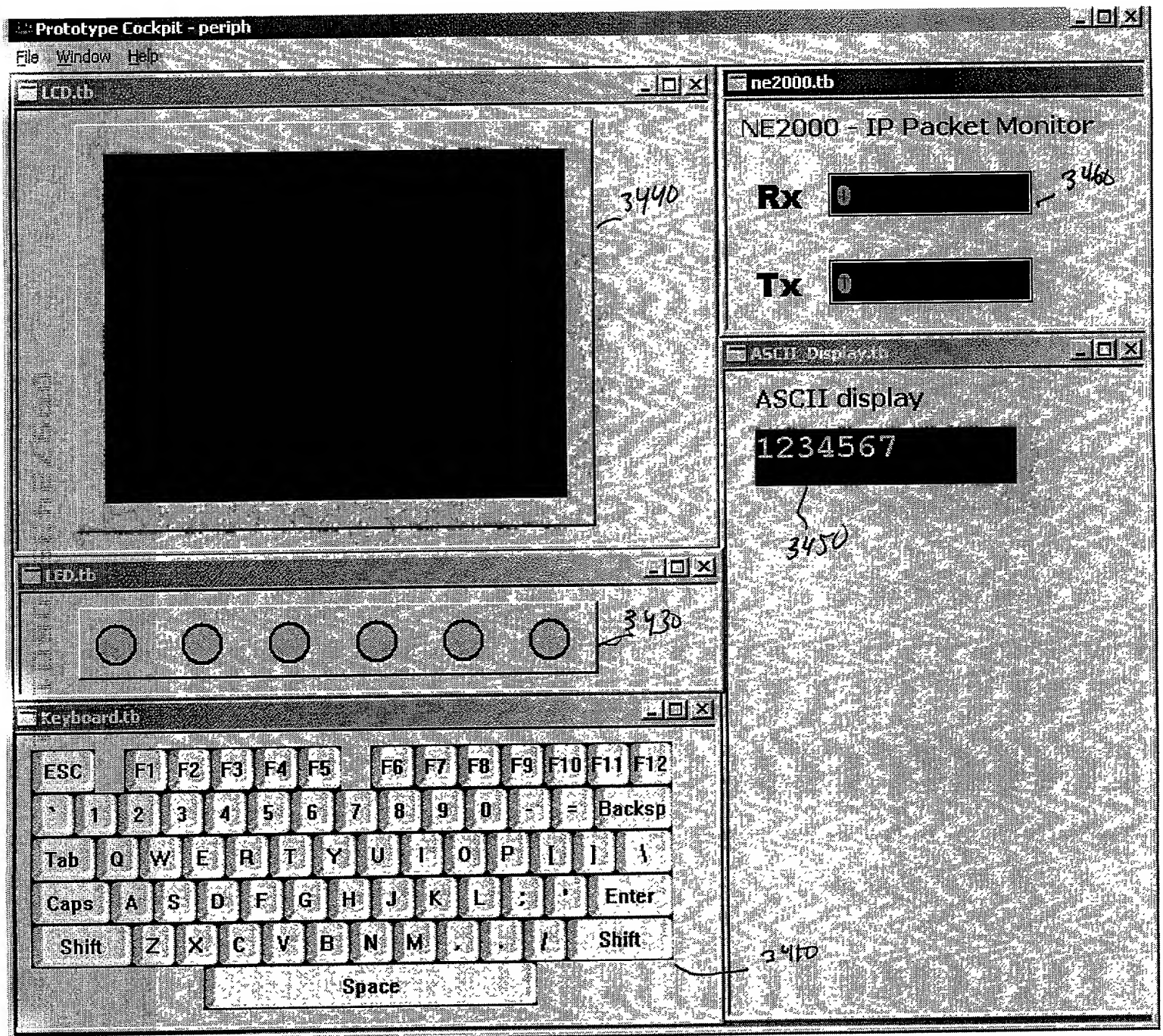


FIG 34B

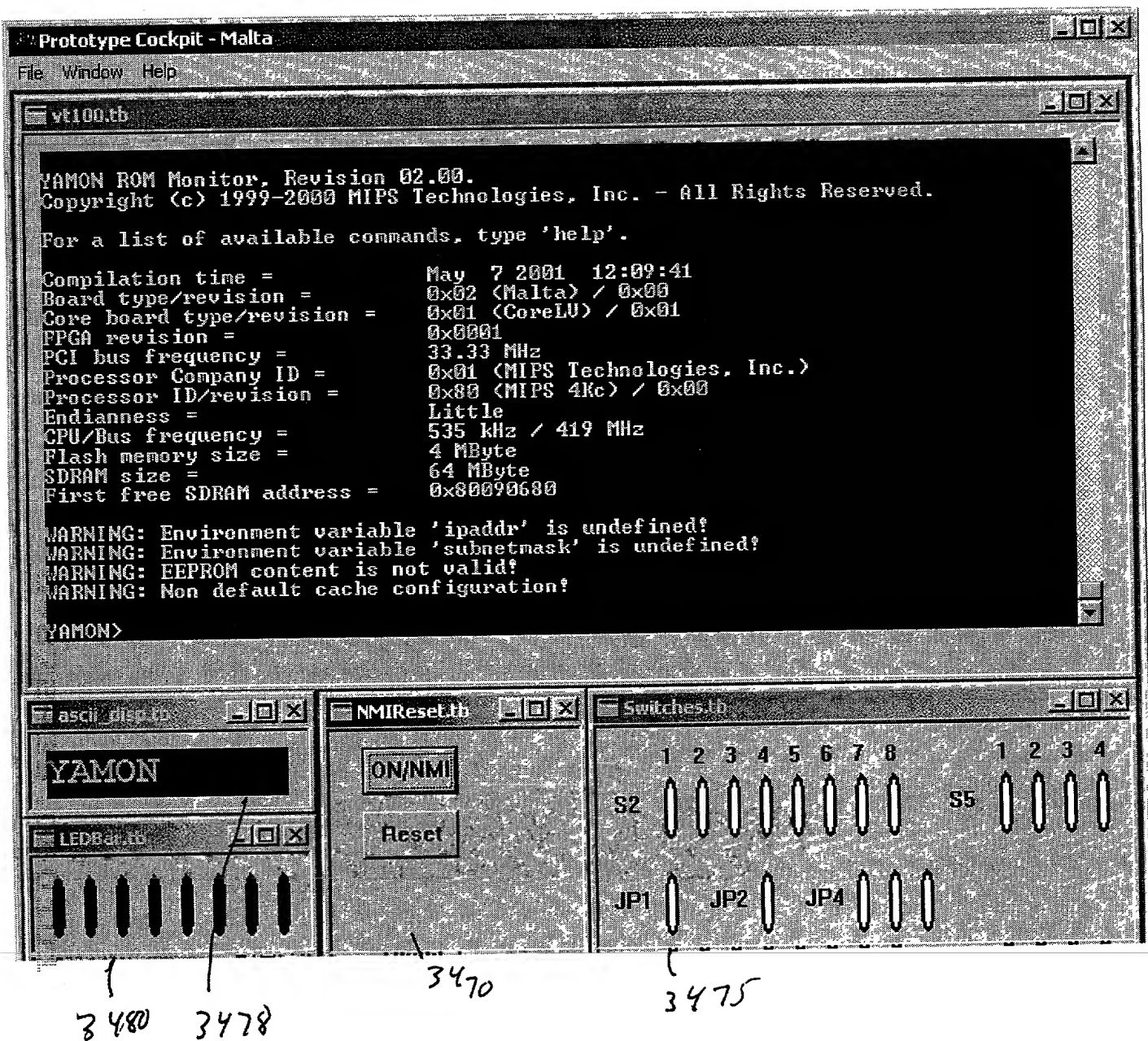


FIG. 34C

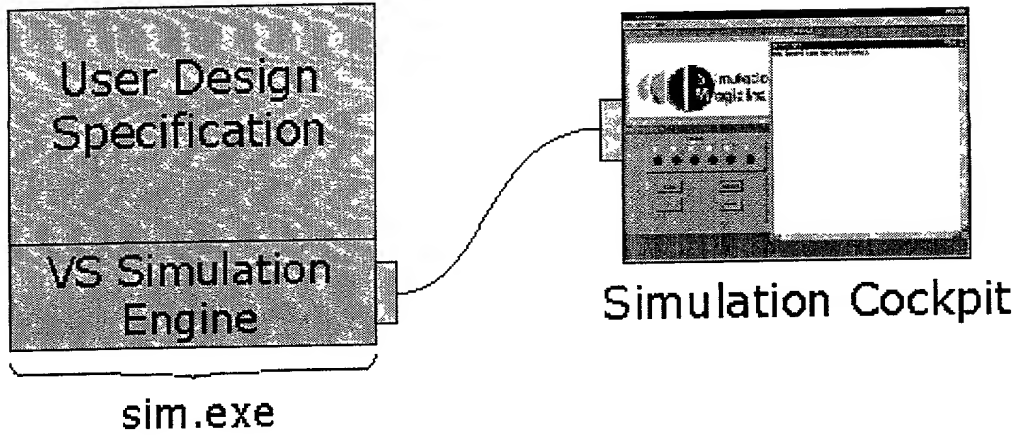


FIG. 35

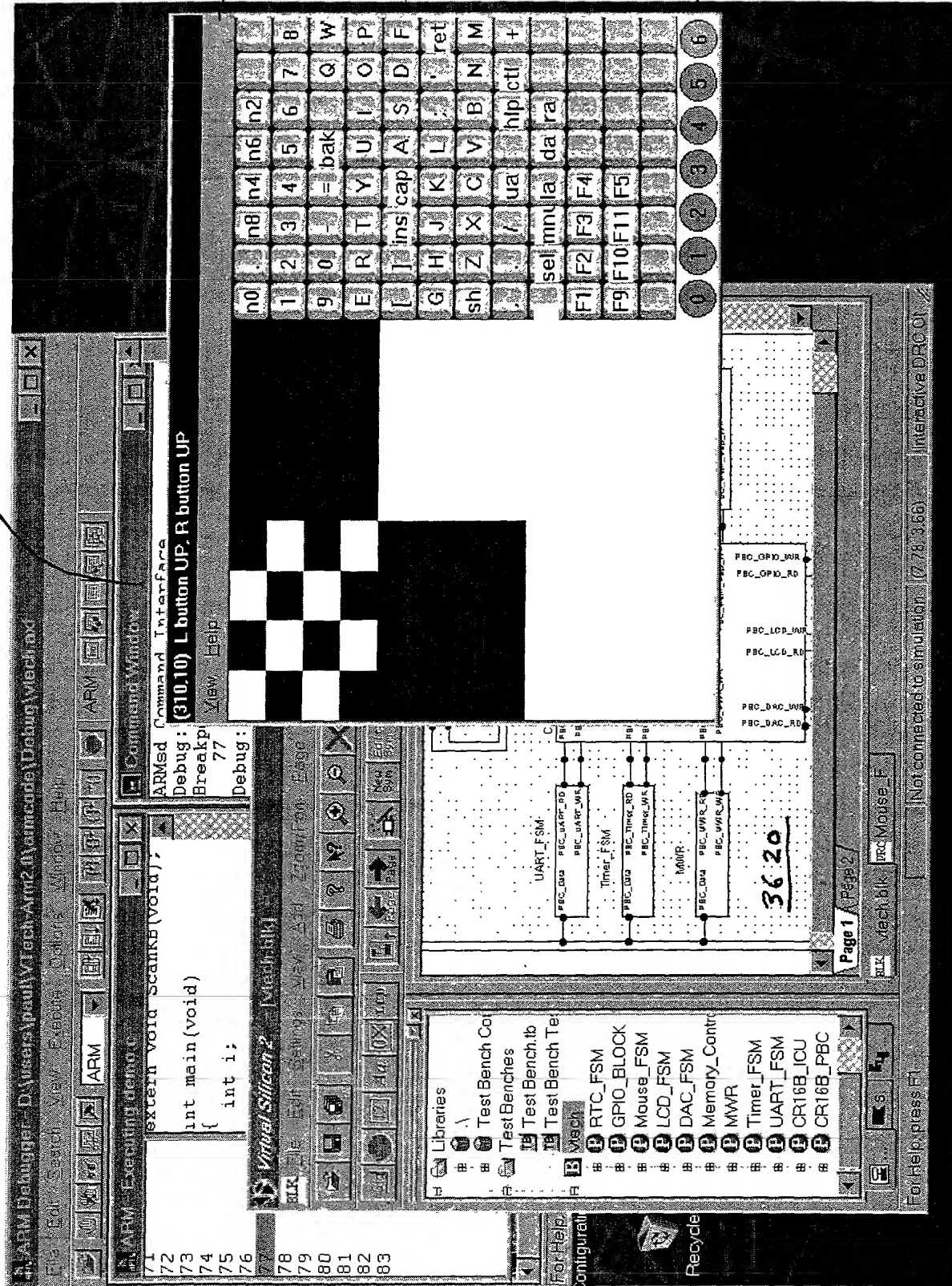
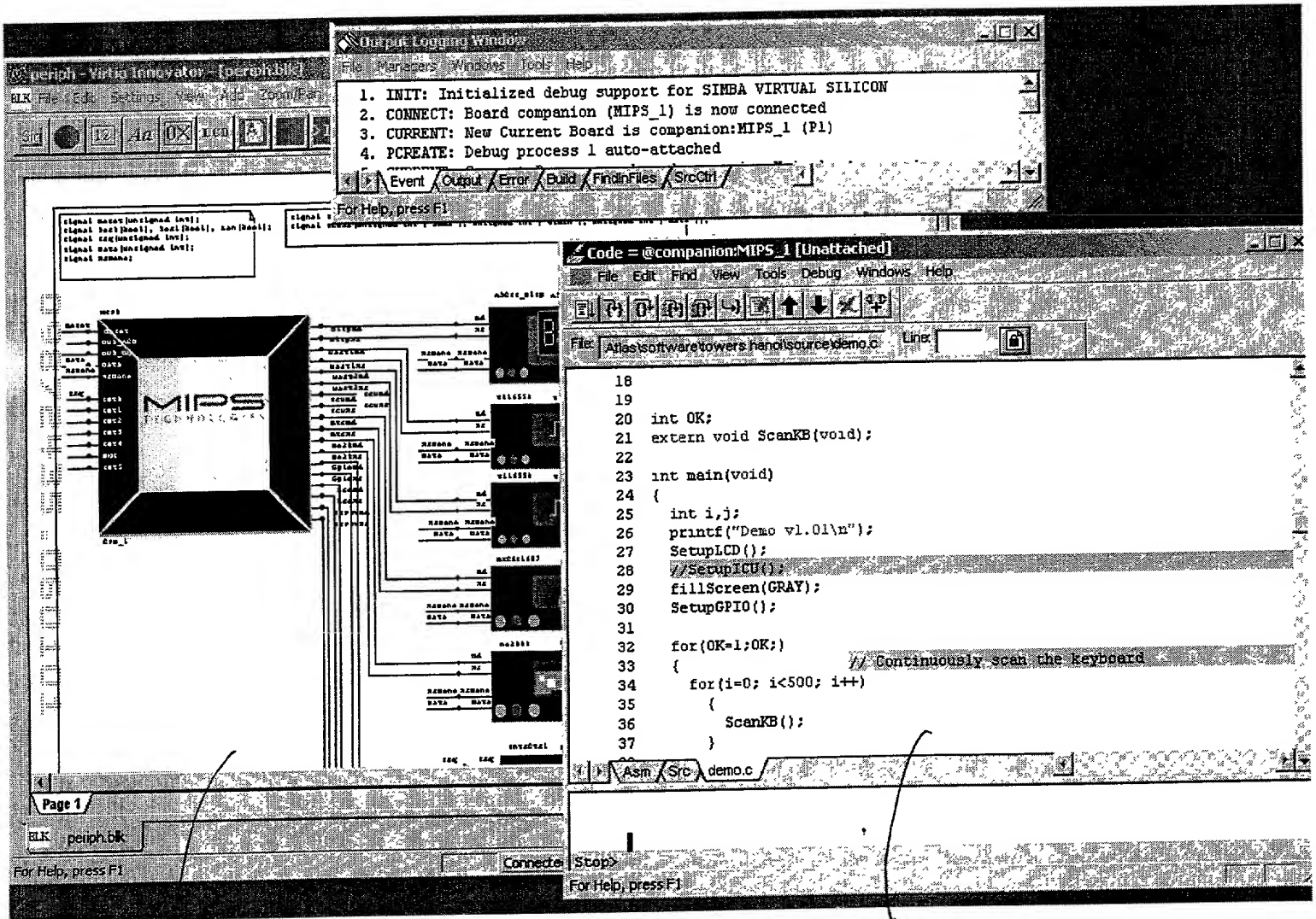


FIG. 36A





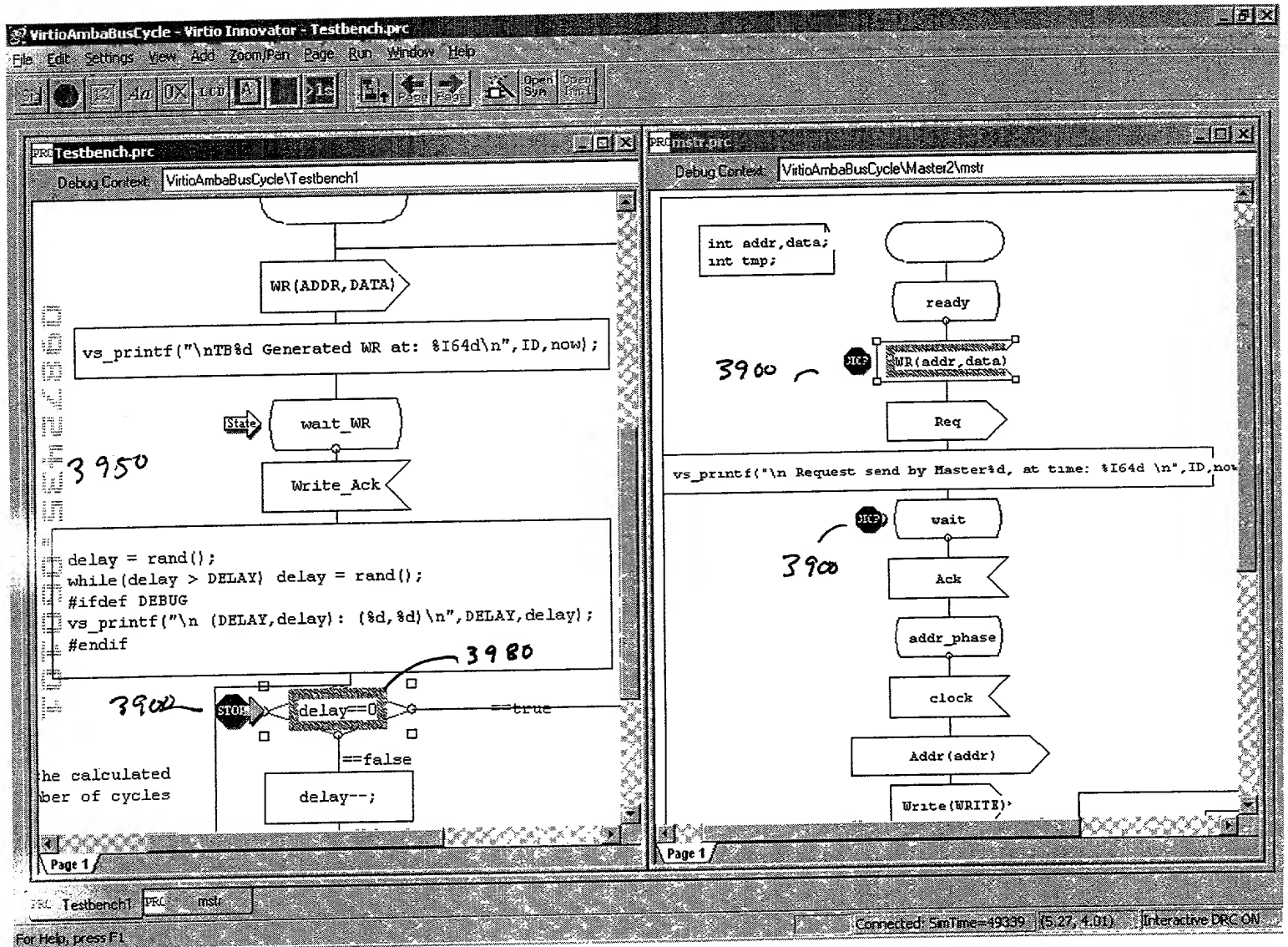


FIG. 36C

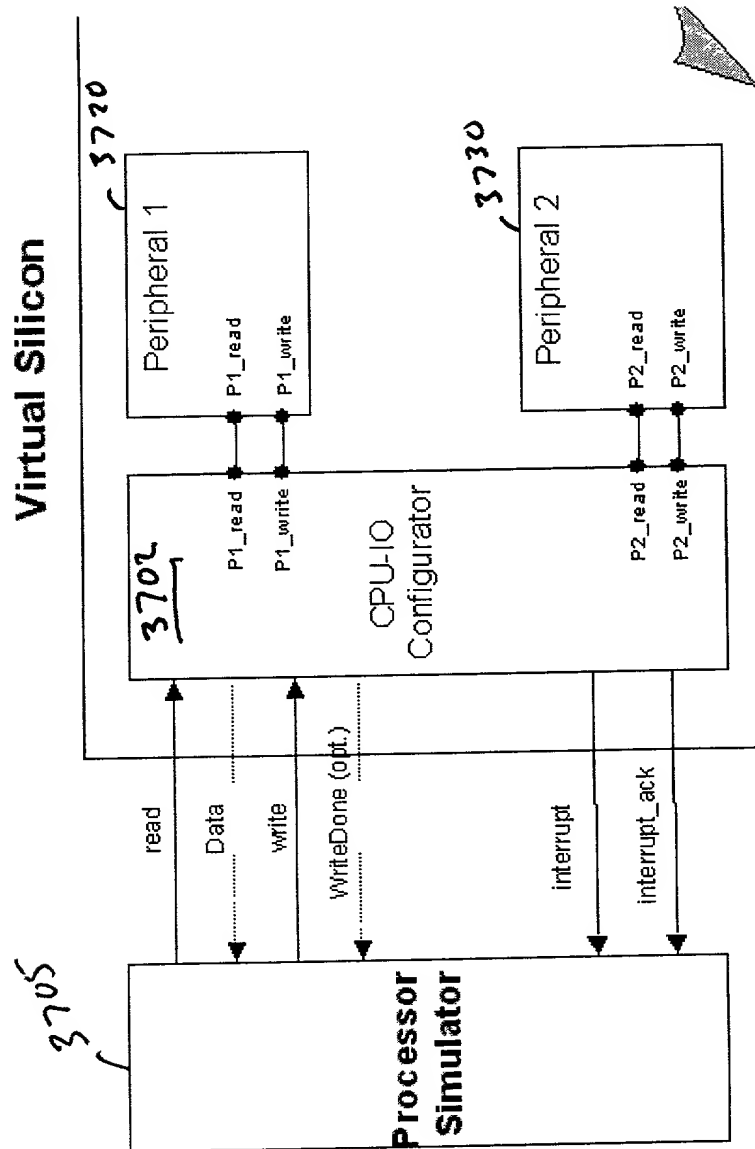


FIG. 37

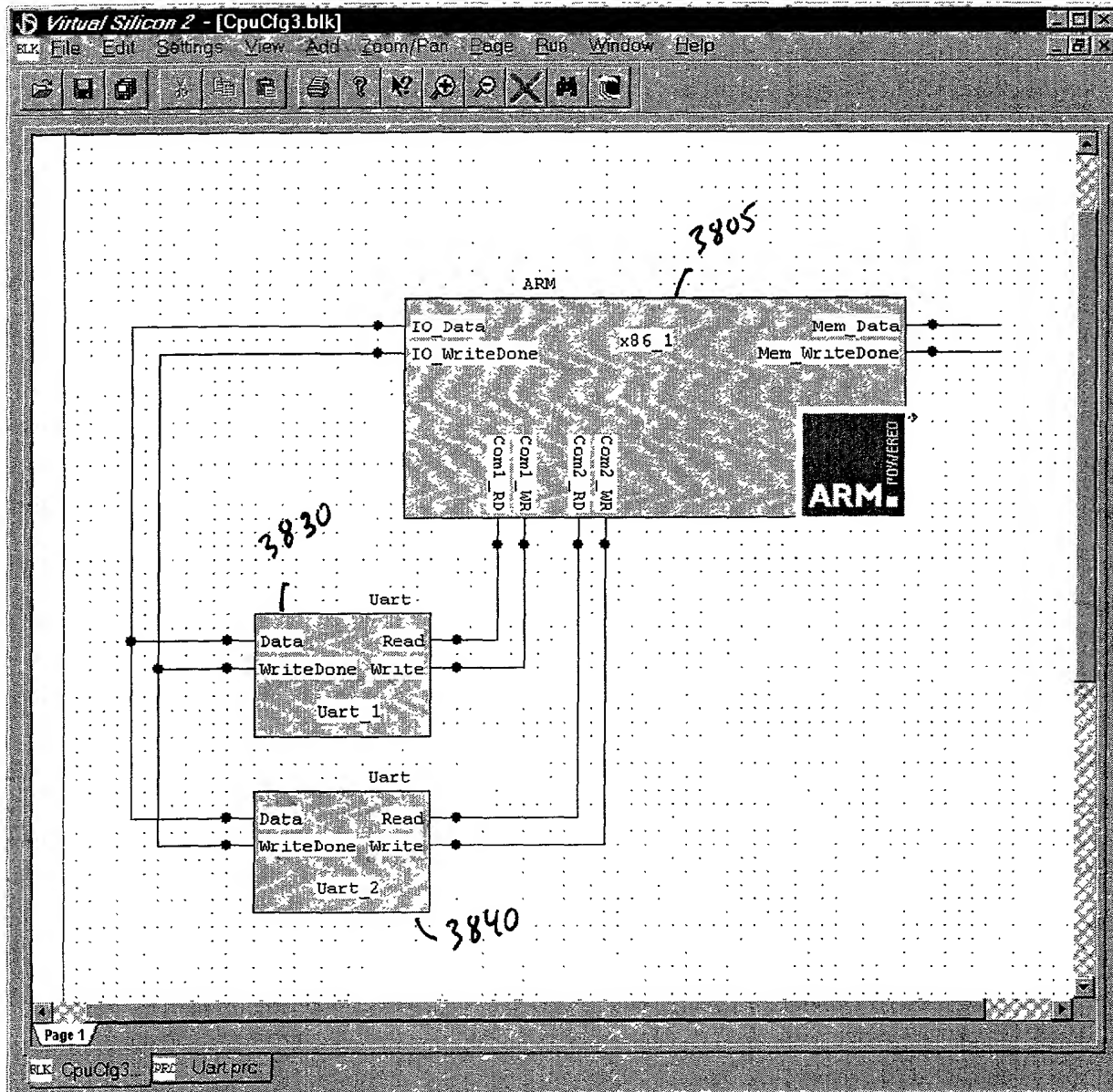


FIG. 38

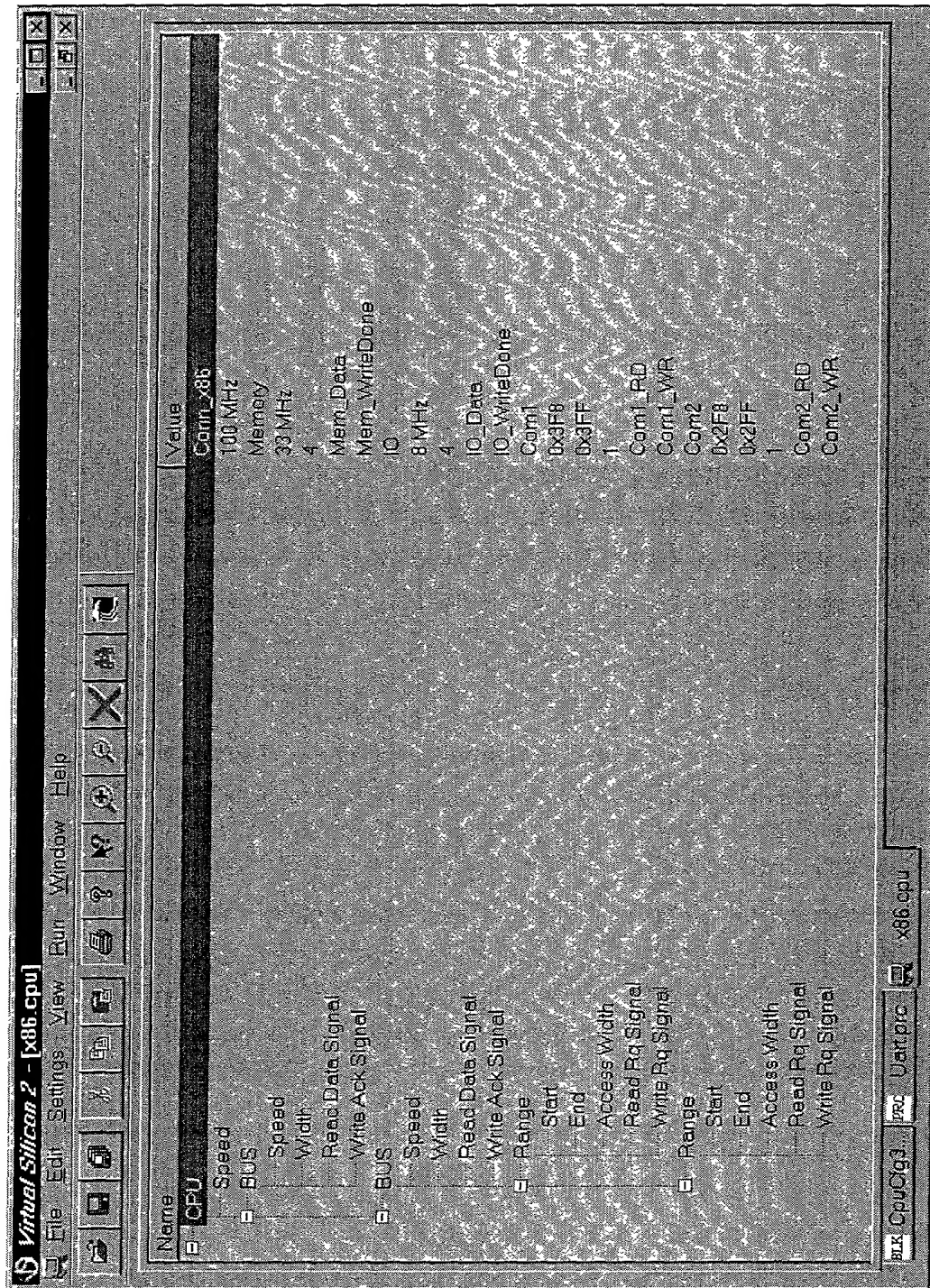


FIG. 39



Name	Value
<input checked="" type="checkbox"/> CPU	ArmCPU
Speed	100 MHz
<input checked="" type="checkbox"/> Events: CPU -> Simulation	
RESET Signal	Reset
BUS_ACK Signal	Bus_ACK
<input checked="" type="checkbox"/> Events: Simulation -> CPU	
BUS_RD	Bus_RD
<input checked="" type="checkbox"/> Interrupt Support	Yes
FIQ Signal	FIQ
IRQ Signal	IRQ
<input checked="" type="checkbox"/> BUS	Memory
Speed	100 MHz
Width	4
Read Data Signal	Read_Data
<input checked="" type="checkbox"/> Write Timing	Variable
Write Ack Signal	Write_Ack
<input checked="" type="checkbox"/> Range	Counter
Type	Slave
Start	CNT_START_ADDR
End	CNT_END_ADDR
Access Width	4
Read Rq Signal	CNT_RD
Write Rq Signal	CNT_WR

FIG. 40

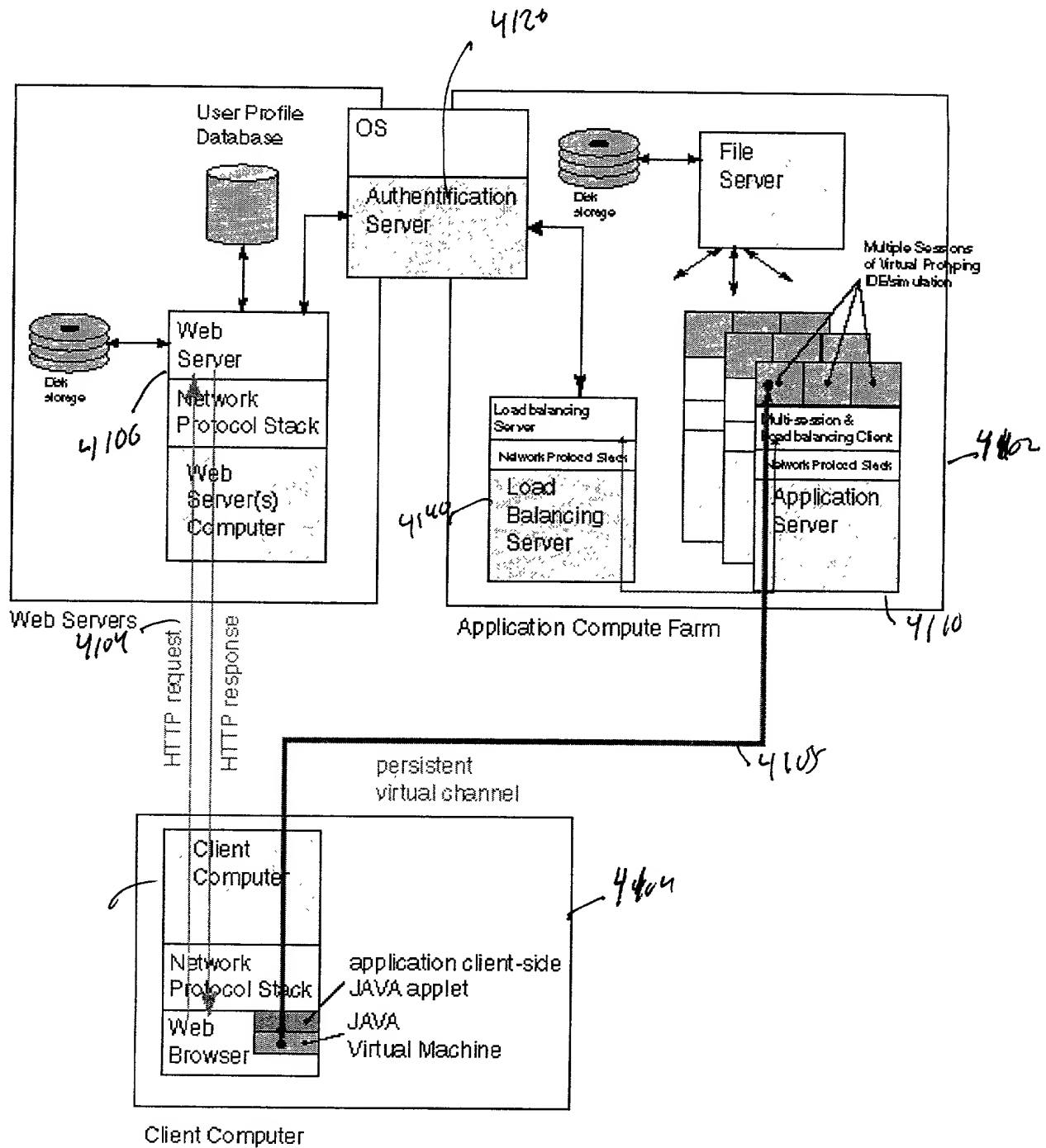
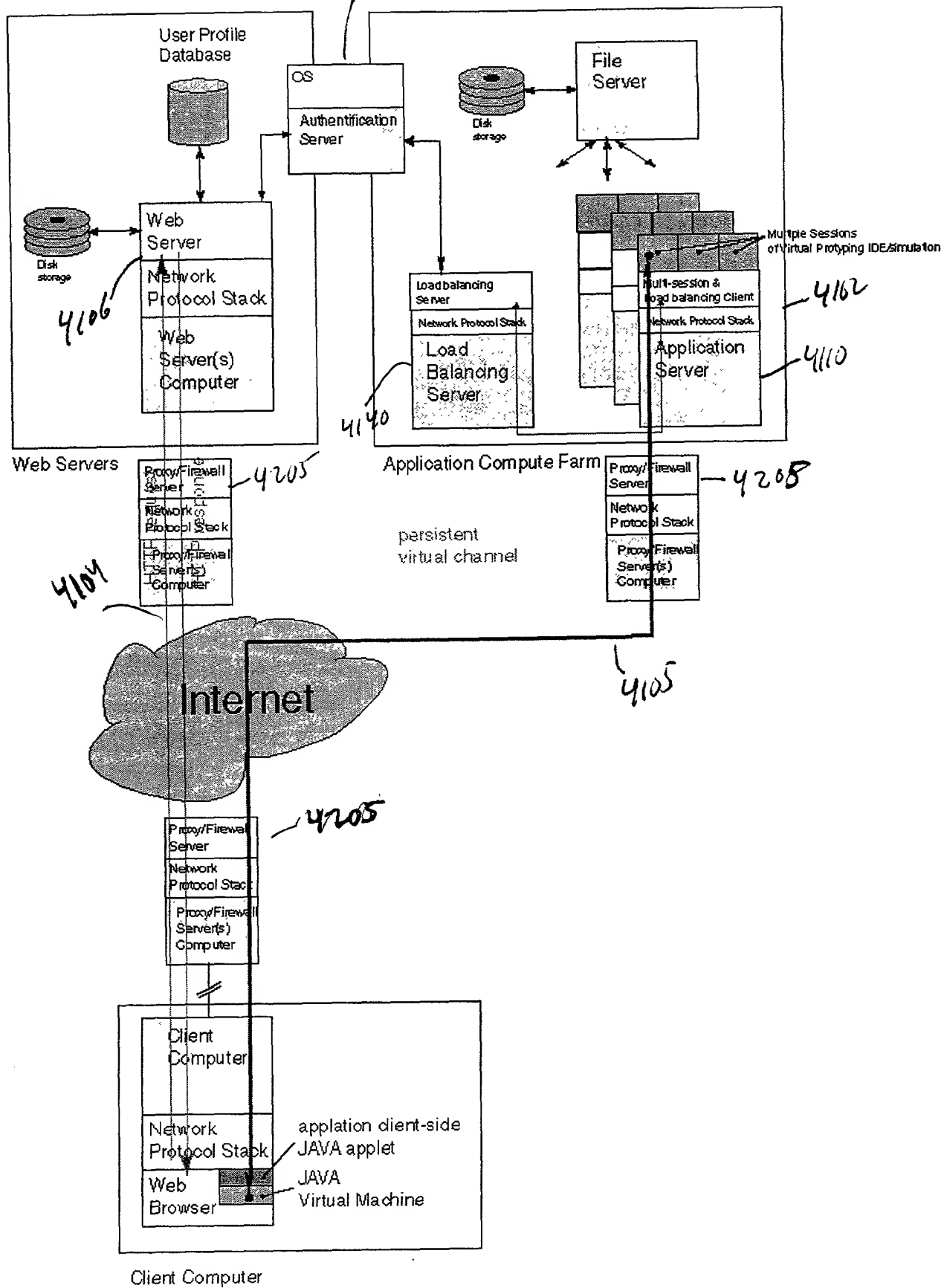


FIG. 41





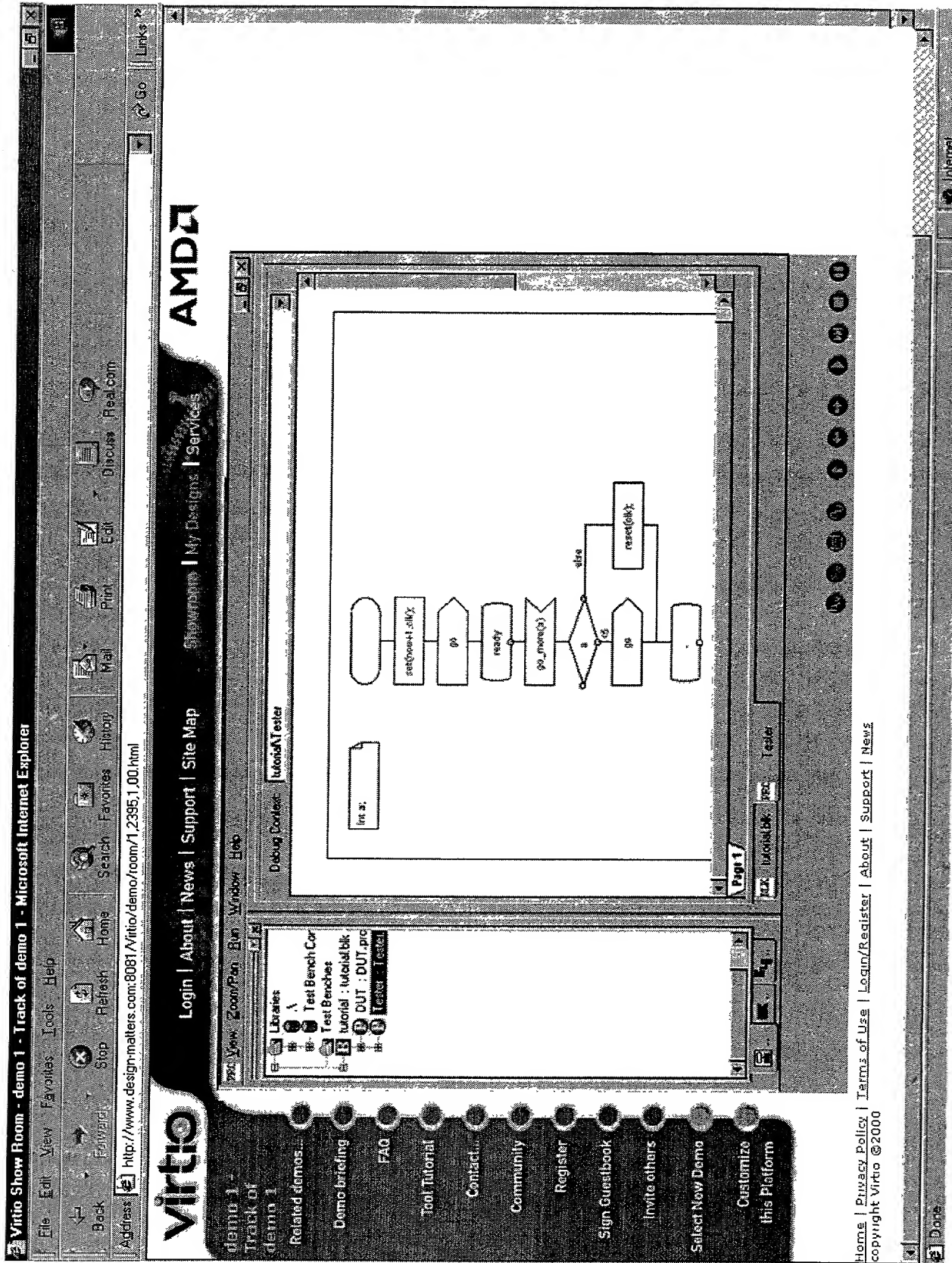


FIG-43

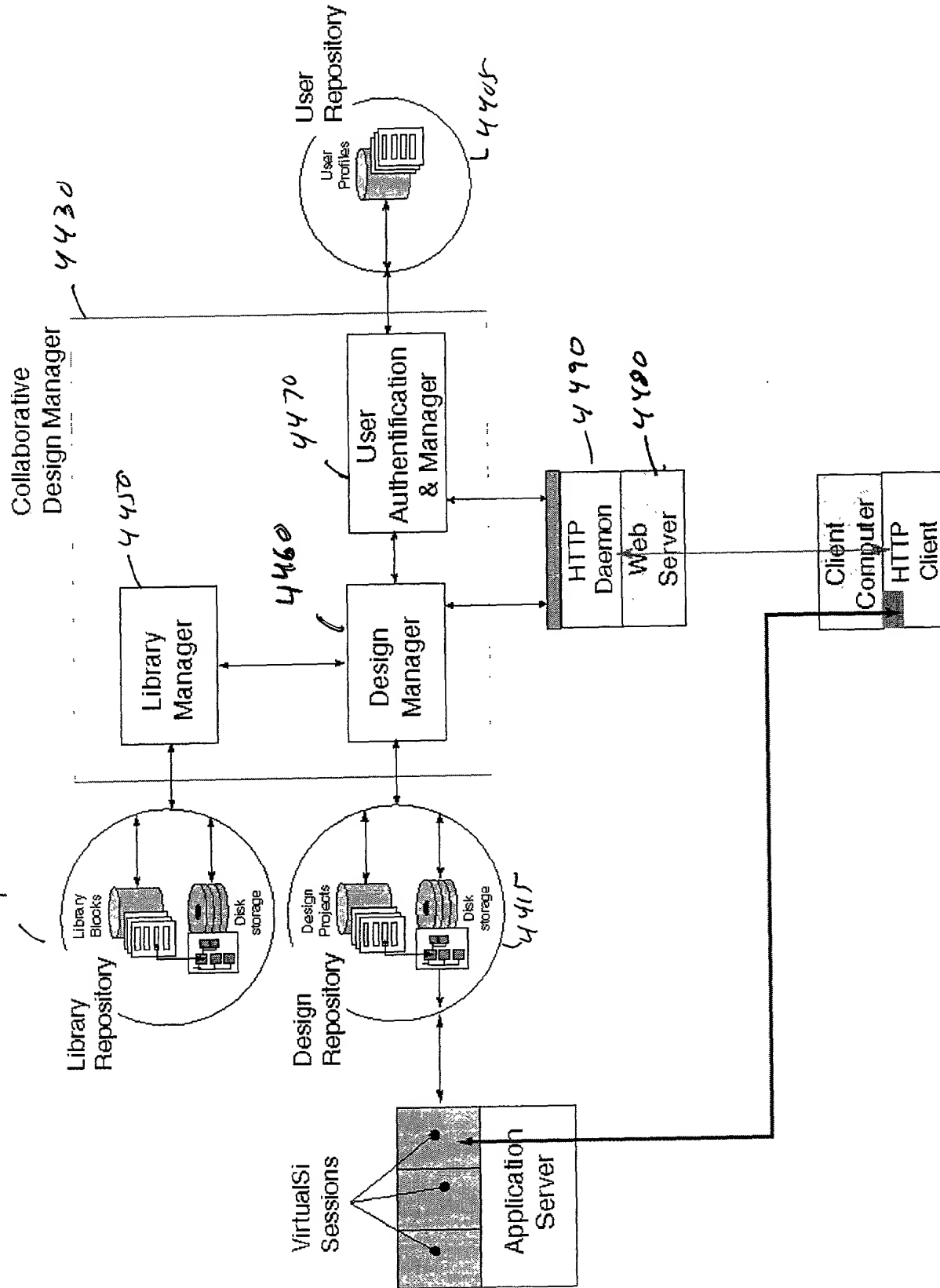


FIG. 44









 Clock_1	Scope name.
 clk(int)	Signal name and declaration.
 t	Timer or clock name and declaration.
 clk	Local name of a signal coming from the upper scope (inherited).
 myClk	Timer or clock is being set.
	Signal is being sent.
	Signal is being received.
	Signal is being saved.

FIG. 45

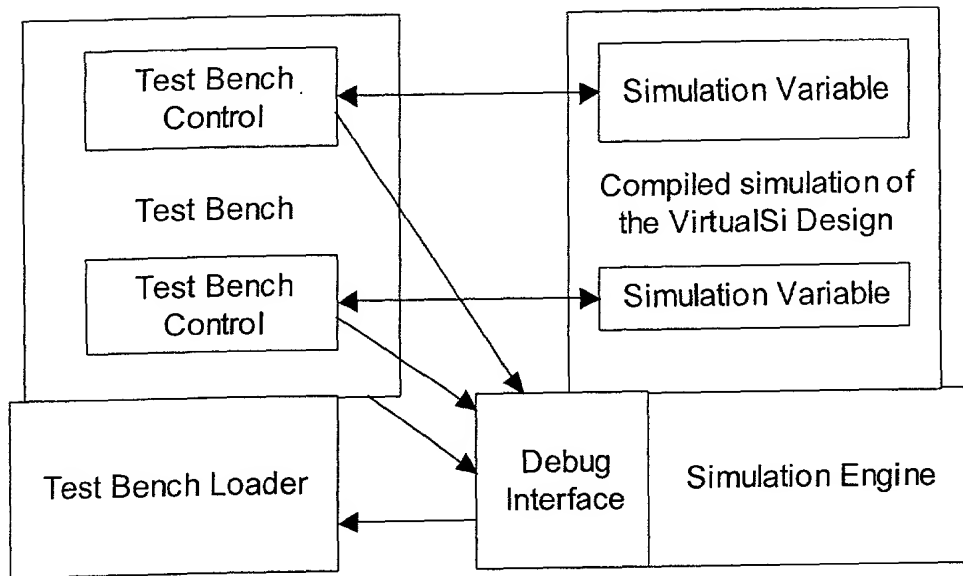


FIG. 46

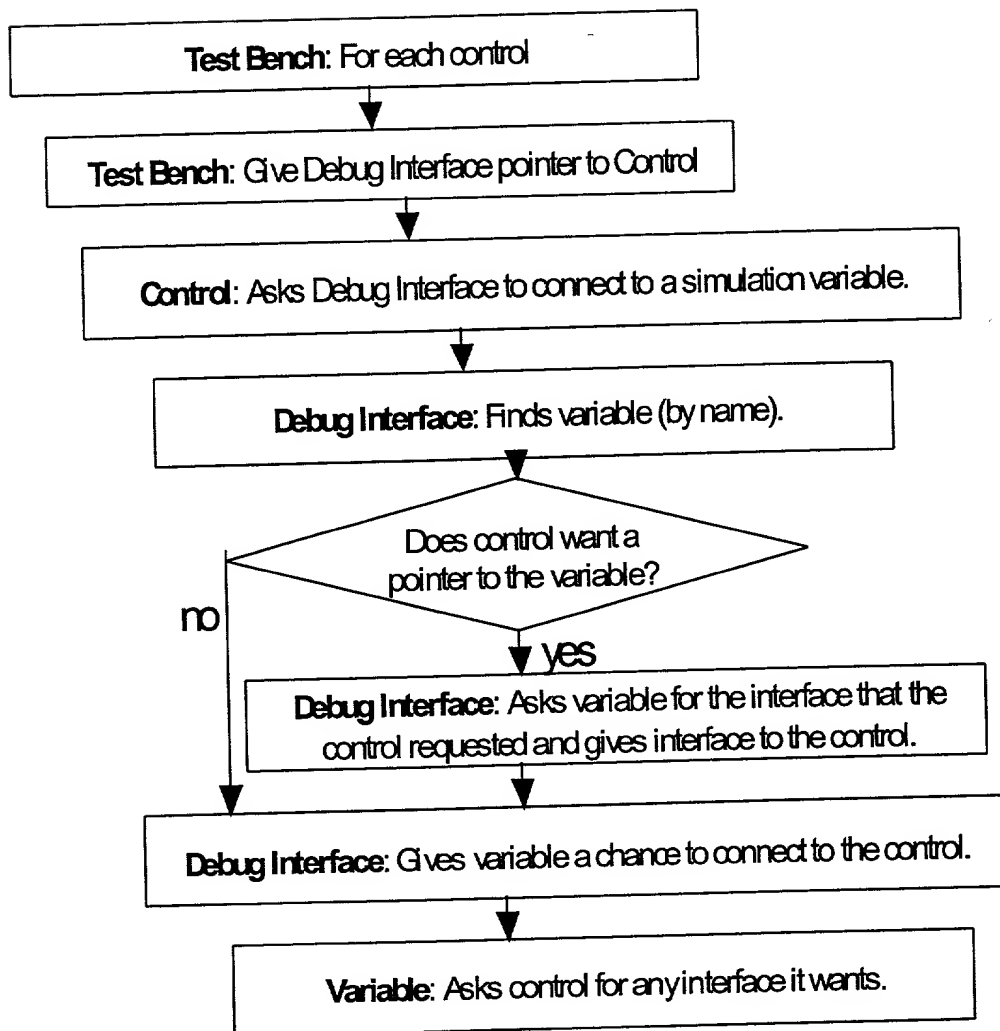


FIG. 47

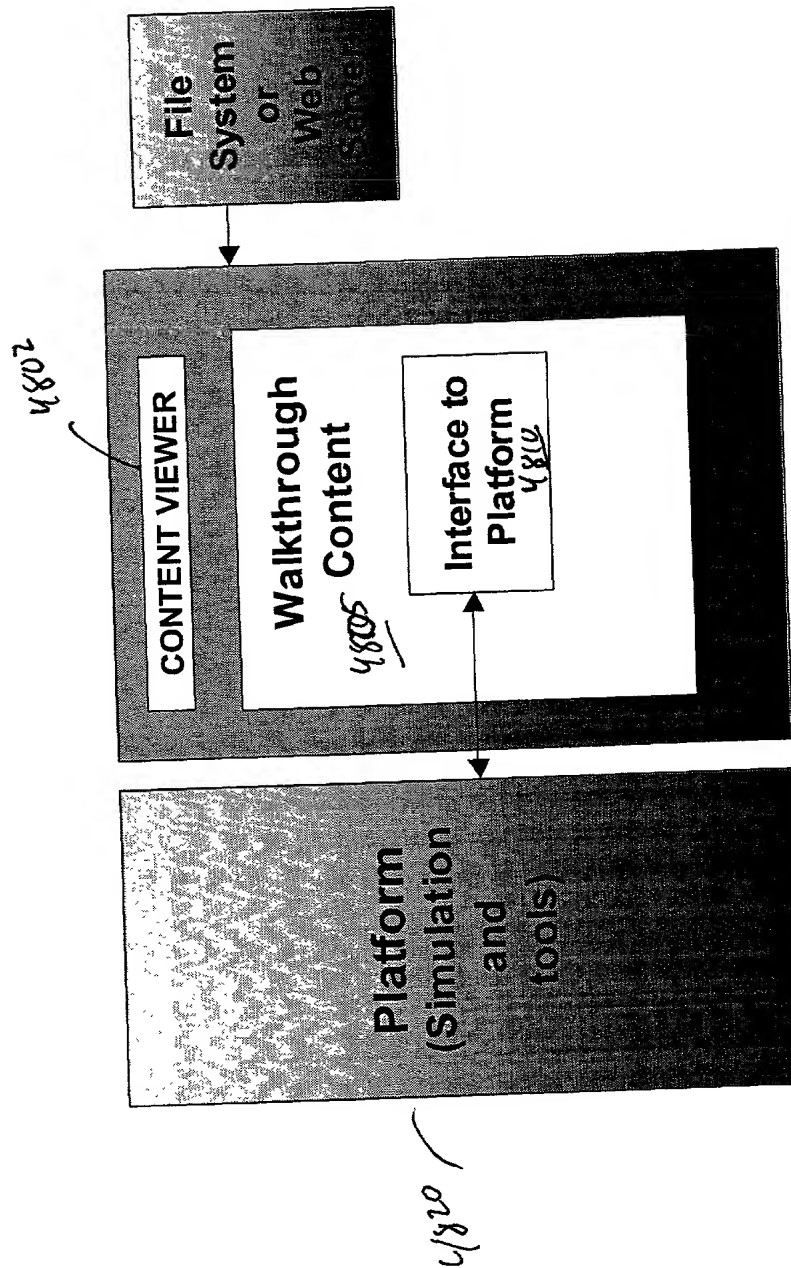


FIG. 48



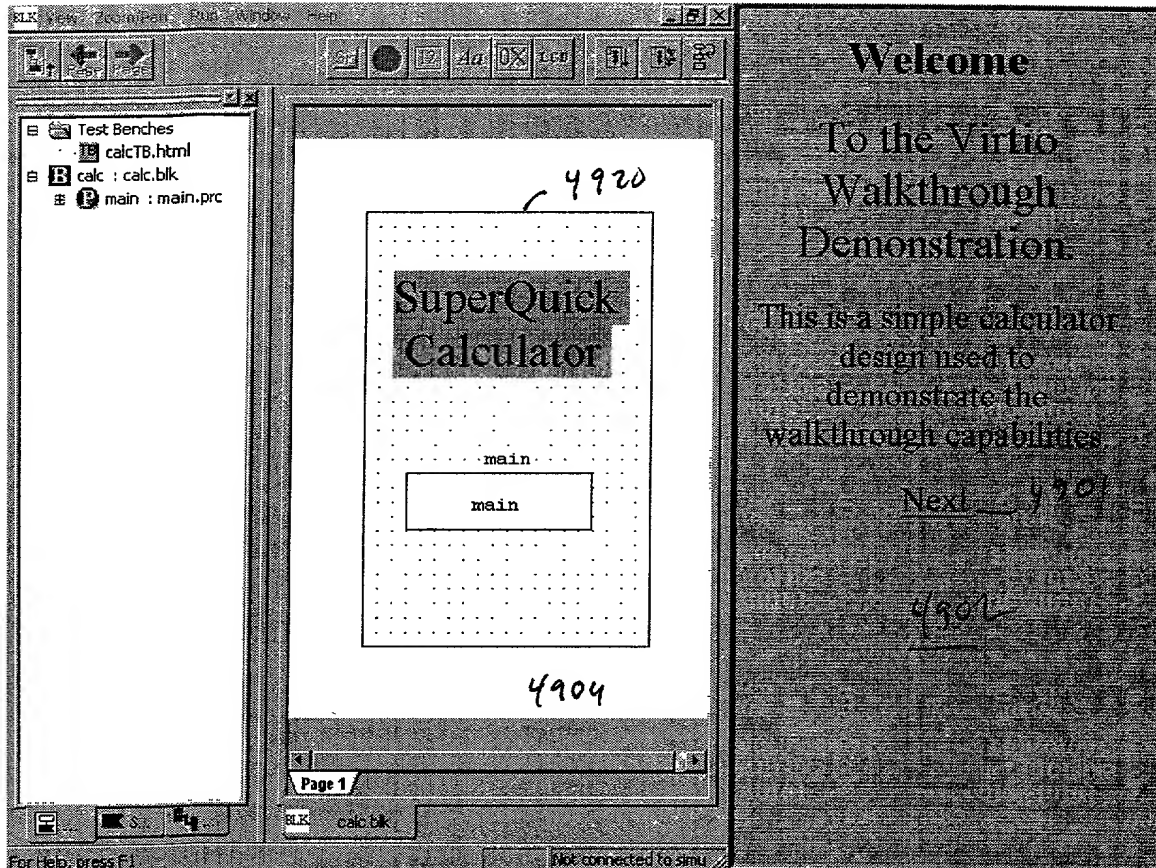
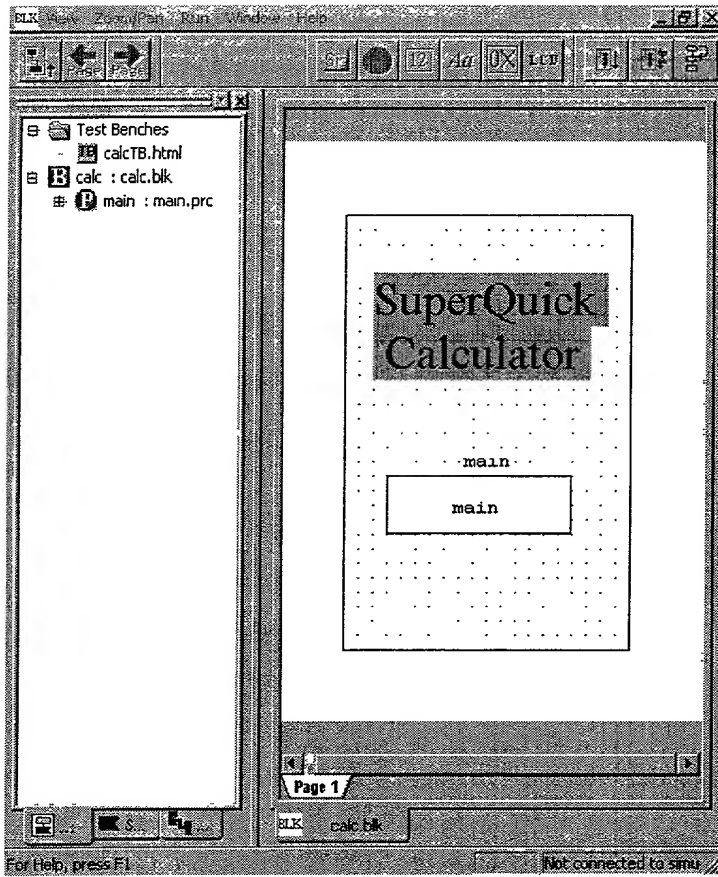


FIG. 49A



## The Design

To the left you see the  
**Virtio Viewer**  
 First let's explore the design.

To make more room on the screen,  
 first close the browser's window  
 (Show me - second click removes  
 box) (Close it)

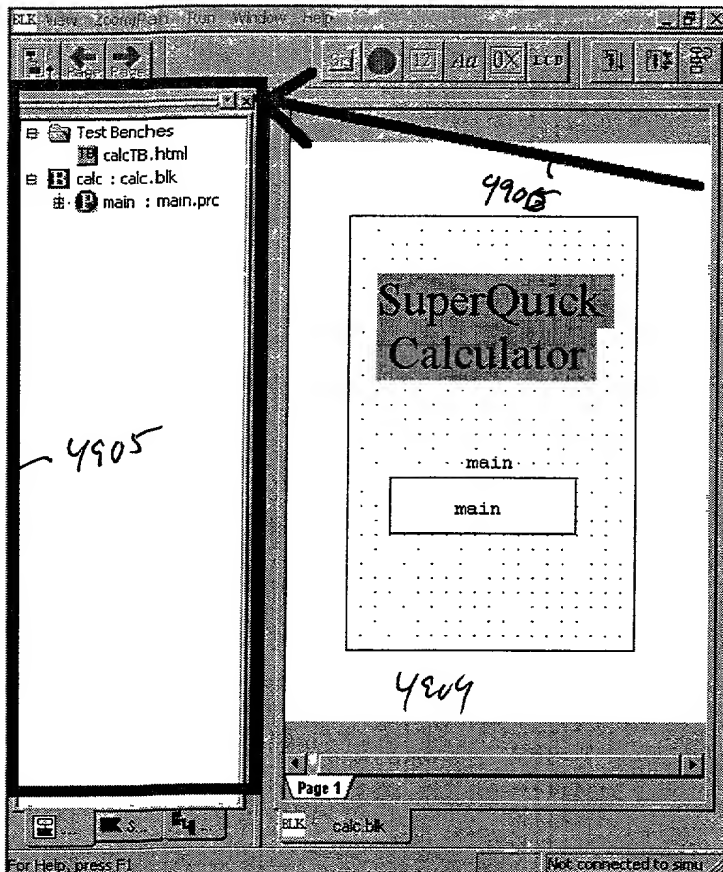
The top level diagram (show  
 diagram) only has one symbol  
 (show symbol)

Double click on the symbol or click  
 here to show the symbol's  
 implementation. This is a process  
 diagram (state machine) 3, that  
 does all the work for this design.

Previous Next

4903

FIG. 49B



## The Design

To the left you see the  
**Virtio Viewer**.  
 First let's explore the design.

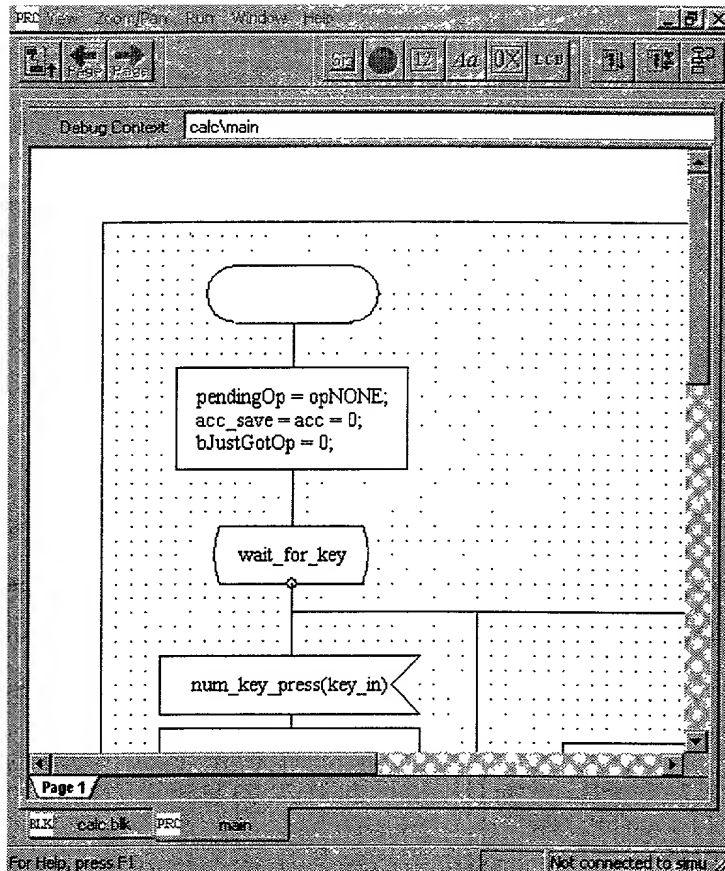
To make more room on the screen,  
 first close the browser's window.  
 (Show me - second click removes  
 box) (Close it)

The top level diagram (show  
 diagram) only has one symbol  
 (show symbol)

Double click on the symbol or click  
 here to show the symbol's  
 implementation. This is a process  
 diagram (state machine) 3. that  
 does all the work for this design.

[Previous](#) ... [Next](#)

FIG. 495



## The Design

To the left you see the  
**Virtio Viewer**  
 First let's explore the design.

To make more room on the screen,  
 first close the browsers window  
 (Show me - second click removes  
 box) (Close it)

The top level diagram (show  
 diagram) only has one symbol  
 (show symbol).

Double click on the symbol or click  
 here to show the symbol's  
 implementation. This is a process  
 diagram (state machine) 3. that  
 does all the work for this design.

Previous ... Next

FIG. 49 D

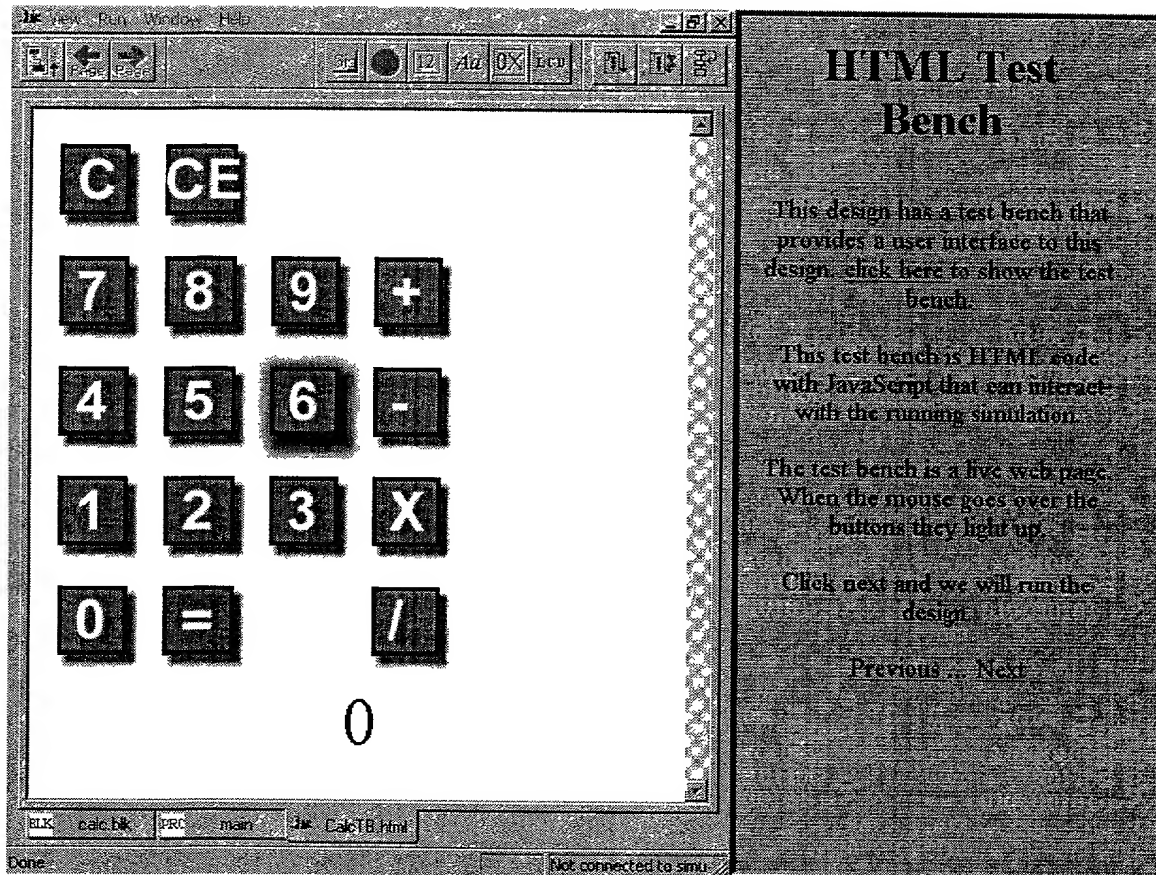
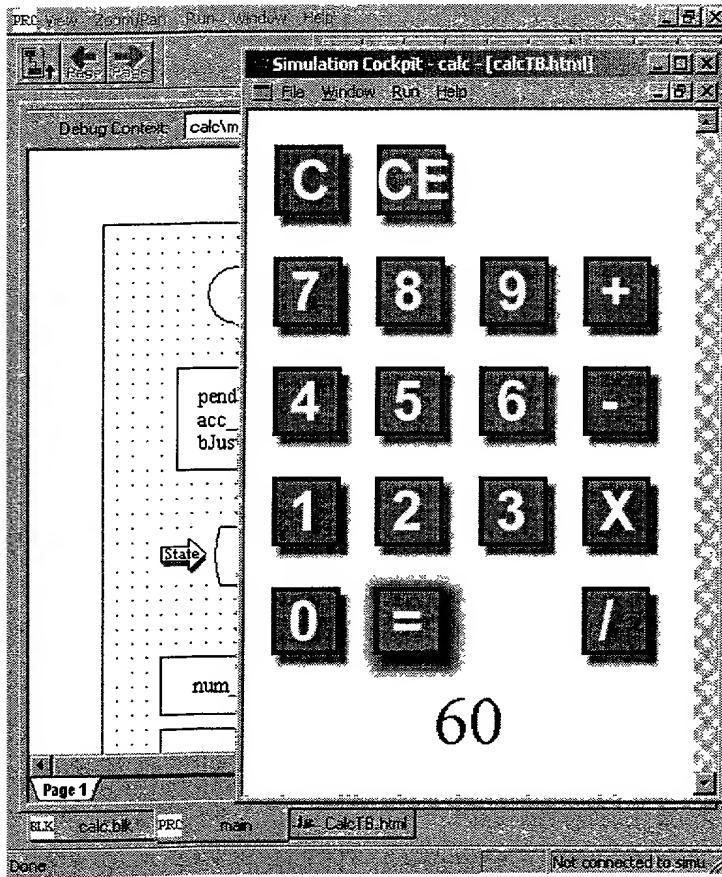


FIG. 49E



## Running the Simulation

Press the arrow button to start the simulation.



The simulation is a separate compiled executable, so it runs very fast. The simulation displays the test bench which now acts as a user interface for the simulation.

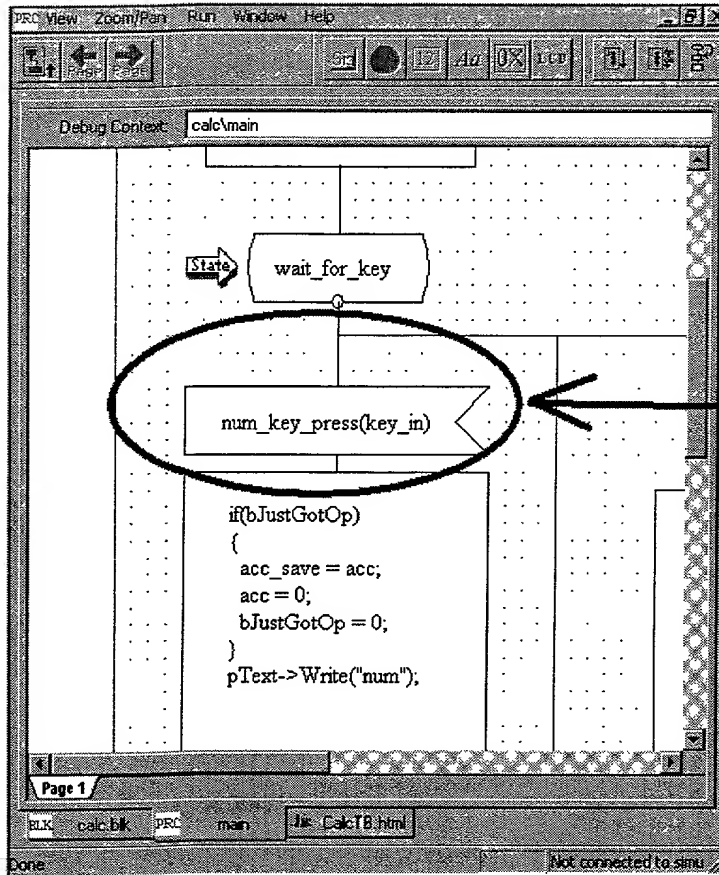
Try clicking on some of the buttons and watching the number on the bottom.

Click next and we will set a breakpoint.

[Previous](#) [Next](#)

FIG. 49F





## Setting a Breakpoint

To set a breakpoint, first click this Viewer icon:



to bring the Virtuo Viewer to the front.

The simulation is still running, it is just behind the viewer now.

We will set a breakpoint on a Signal In construct. Click here to show it. (The next click will remove the arrow).

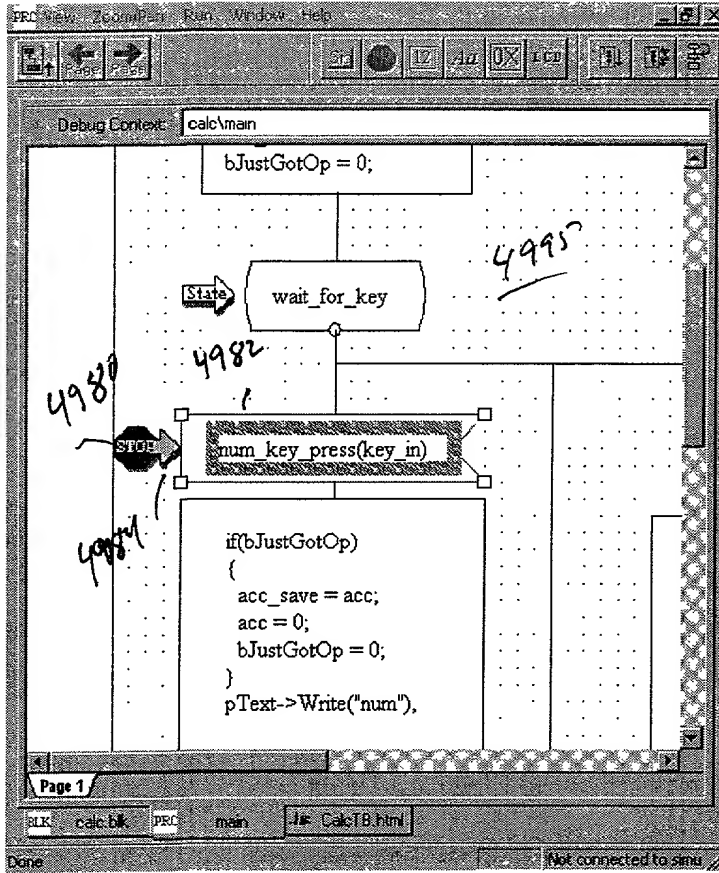
Set a breakpoint by right clicking on the Signal In, or by clicking this Stop icon:




Previous... Next

FIG. 496






## Single Stepping

Click this icon  to bring the simulation to the front again.

Now Click on a number key in the simulation so it will hit our breakpoint.

Click here  to bring the Virtio Viewer forward. It should be stopped at the breakpoint.

You can single step from the Viewer menu or toolbar, or this

icon: 

Single step two times and go to the next page to see how the simulation communicates with the test bench.

[Previous](#) [Next](#)

FJG. 494

## Simulation to Test Bench Communication

Now the simulation should be stopped on a task block. Click here to show it.

A task block is C++ code that is executed as part of a simulation. This task block modifies the variable acc.

On the test bench (click to show it) the number on the bottom displays the current value of acc.

Now step once more while watching the test bench. That is all it takes to update the test bench!

Previous   Next

FIG. 49I

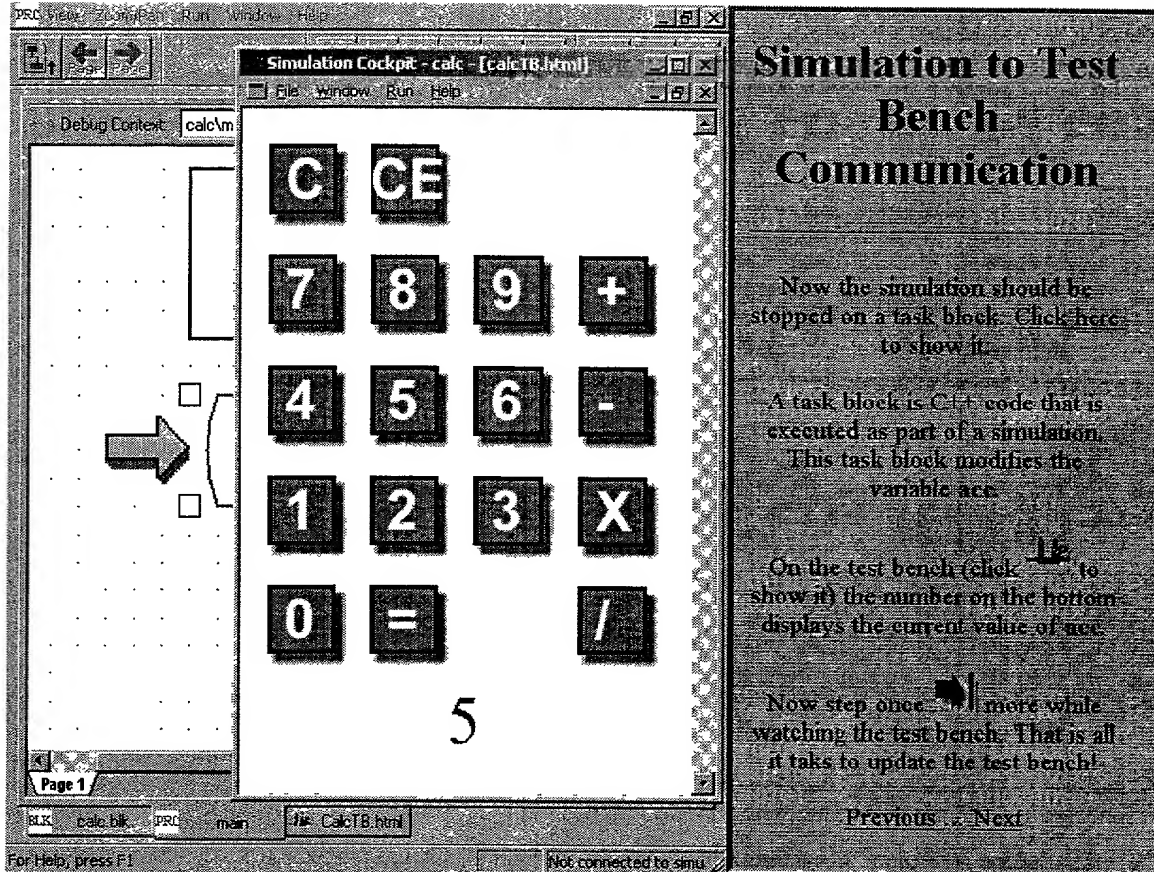
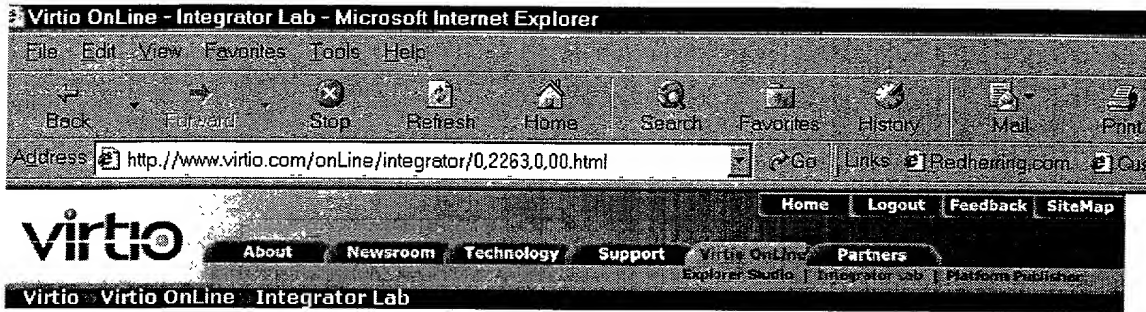


FIG. 49J

## Integrator Lab Screen Snap Shots – On-Line Enablement



### Integrator Lab

Current Designs				
Design Name	Creation Date	Last Edit Date	Description	Delete Project
<a href="#">test</a>	02-May-01	29-May-01	test	<a href="#">Delete</a>
<a href="#">pcnetlink</a>	04-May-01	25-May-01	asdasd	<a href="#">Delete</a>
<a href="#">clonetutorial</a>	23-May-01	23-May-01	testing cloning of build77	<a href="#">Delete</a>
<a href="#">clonehanoi</a>	25-Apr-01	31-May-01	cloning hanoi	<a href="#">Delete</a>
<a href="#">cloneatlas</a>	25-Apr-01	29-May-01	cloning atals	<a href="#">Delete</a>

[New Design](#)

[My Invitees](#)

FIG. 50

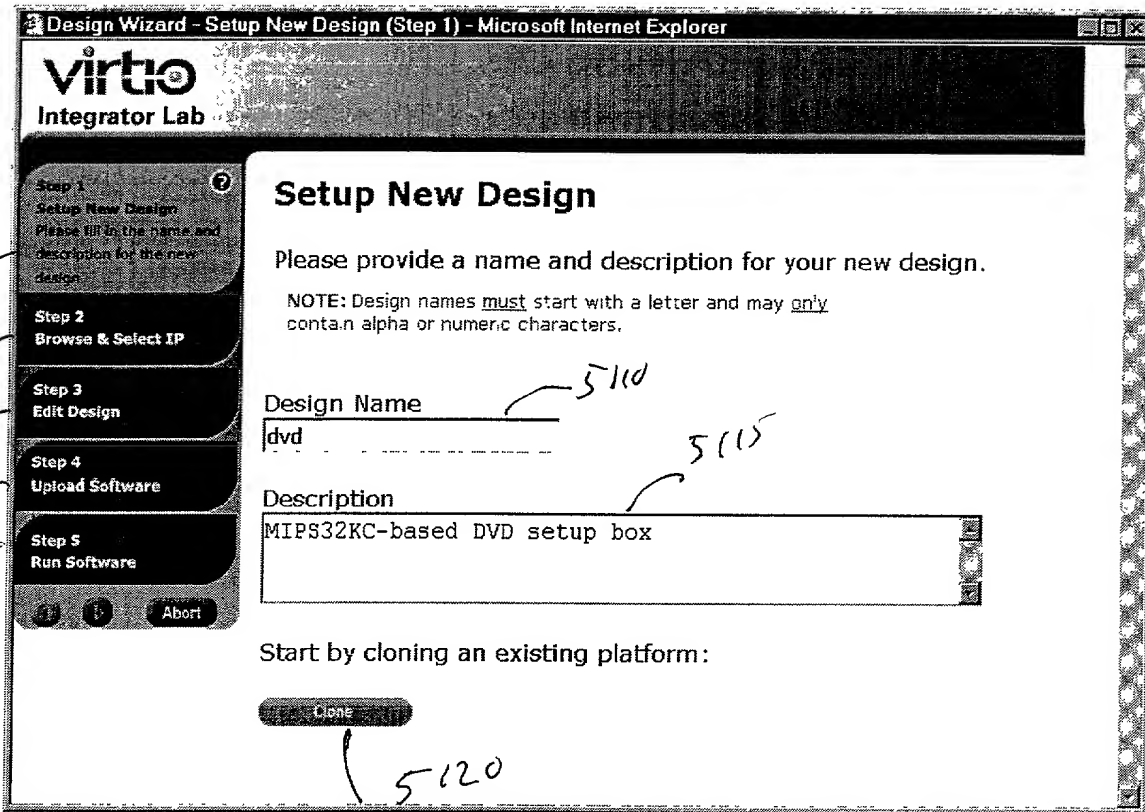


FIG. 51

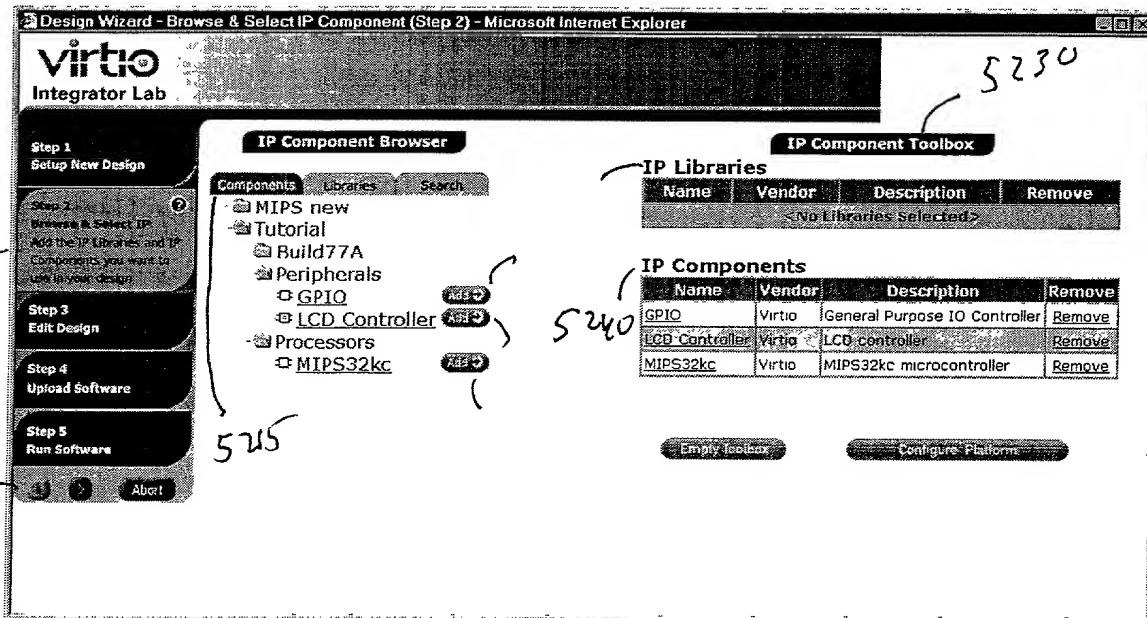


FIG. 52

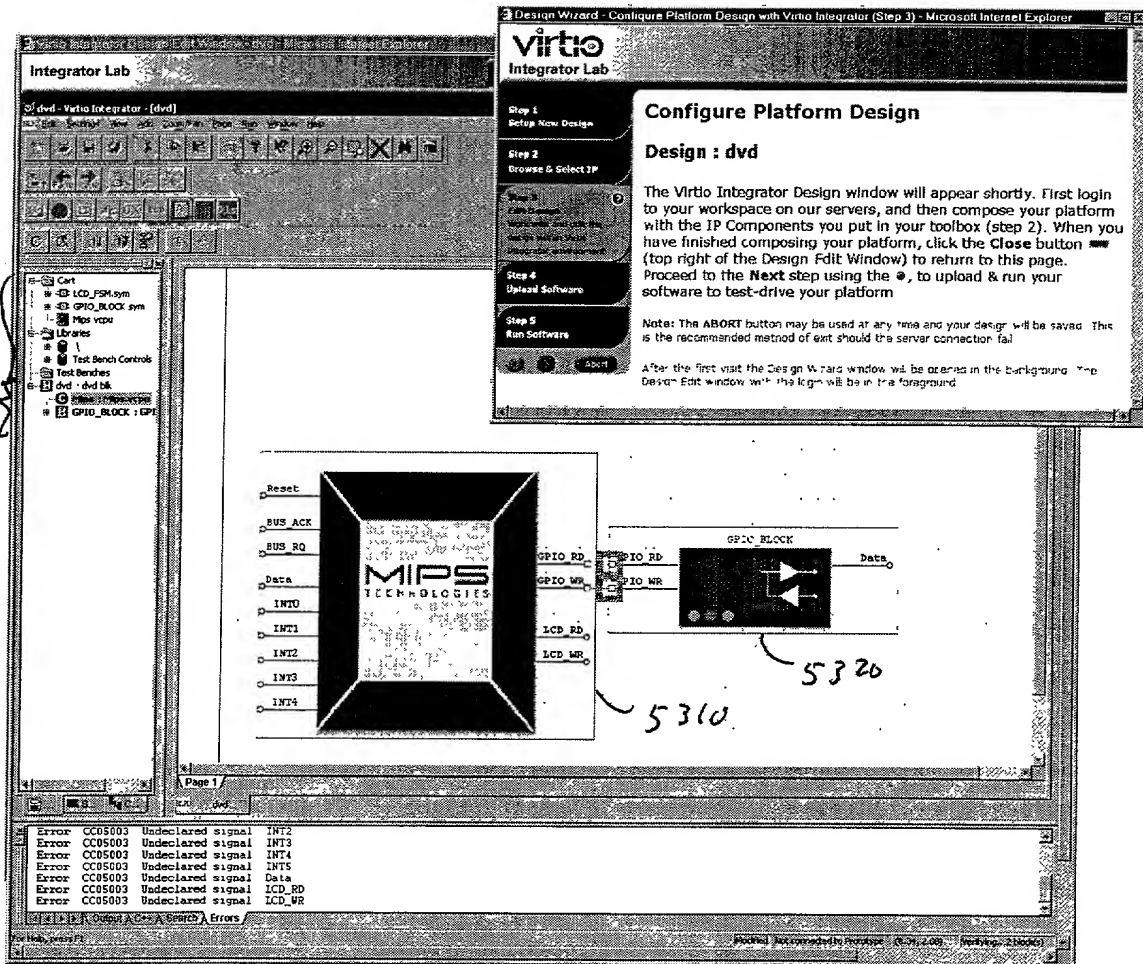


FIG. 53

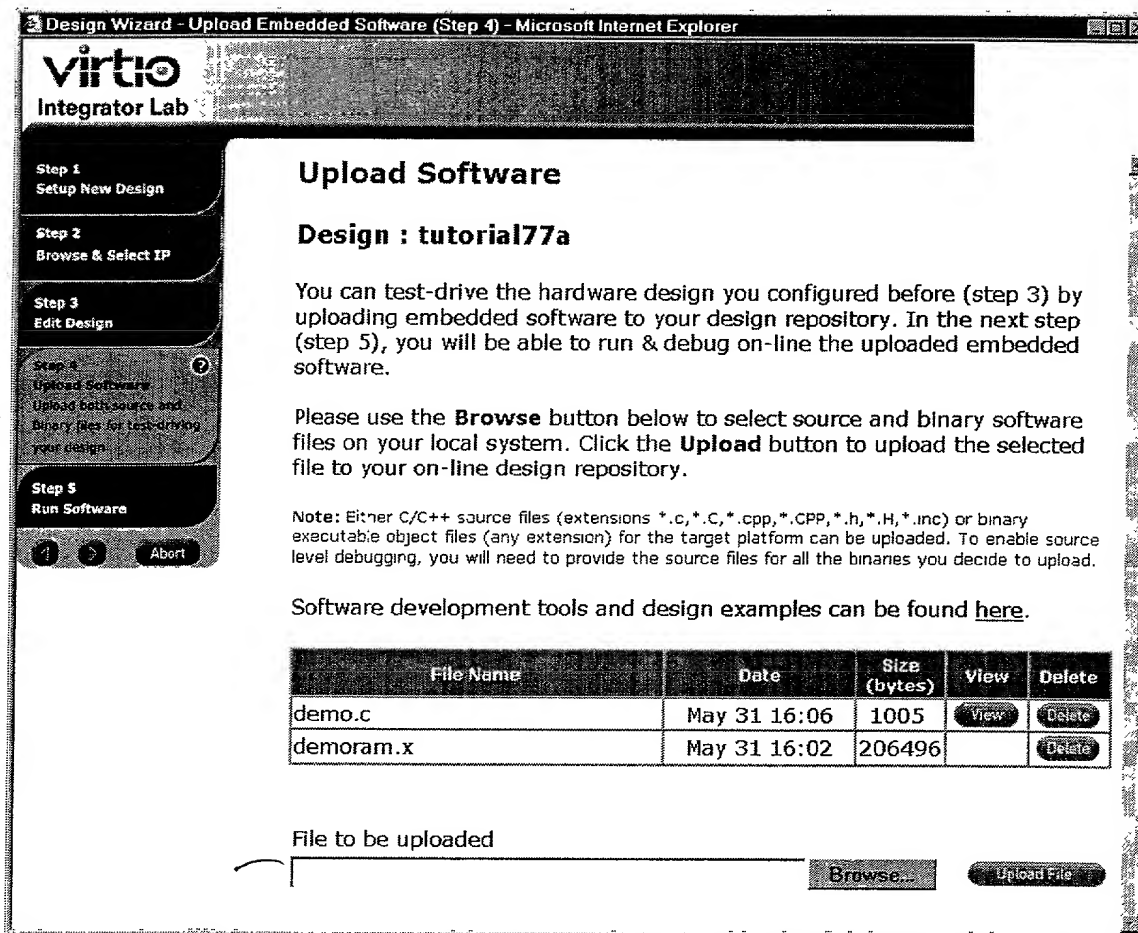


FIG. 54



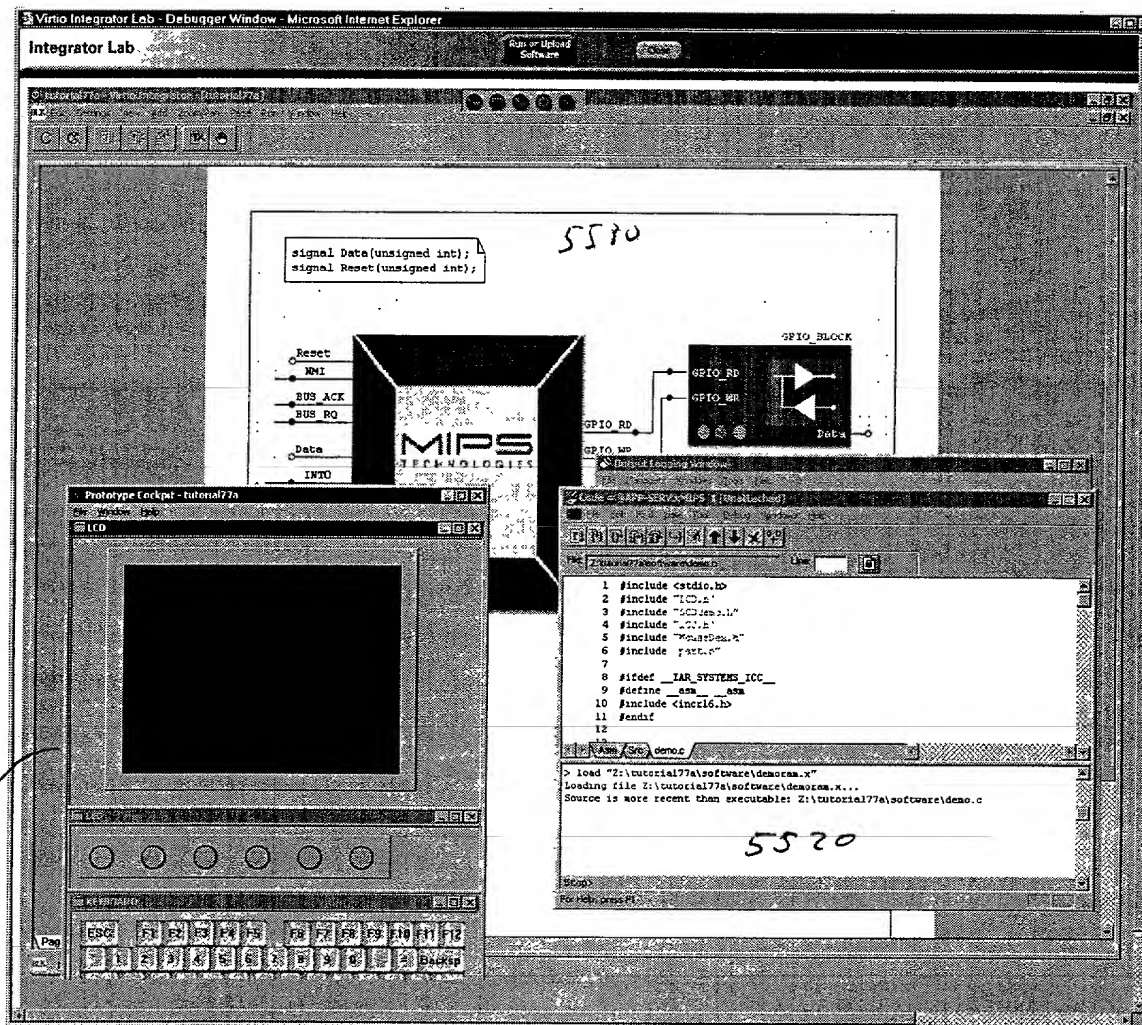


FIG. 55